



Qualcomm Technologies International, Ltd.



Enabling Qualcomm aptX Codecs

Application Note

80-CT446-1 Rev. AG

February 22, 2018

Confidential and Proprietary – Qualcomm Technologies International, Ltd.

NO PUBLIC DISCLOSURE PERMITTED: Please report postings of this document on public servers or websites to DocCtrlAgent@qualcomm.com.

Restricted Distribution: Not to be distributed to anyone who is not an employee of either Qualcomm Technologies International, Ltd. or its affiliated companies without the express approval of Qualcomm Configuration Management.

Not to be used, copied, reproduced, or modified in whole or in part, nor its contents revealed in any manner to others without the express written permission of Qualcomm Technologies International, Ltd.

Qualcomm Kalimba and Qualcomm aptX are products of Qualcomm Technologies International, Ltd. Other Qualcomm products referenced herein are products of Qualcomm Technologies International, Ltd.

Qualcomm is a trademark of Qualcomm Incorporated, registered in the United States and other countries. Kalimba is a trademark of Qualcomm Technologies International, Ltd. aptX is a trademark of Qualcomm Technologies International, Ltd., registered in the United States and other countries. Other product and brand names may be trademarks or registered trademarks of their respective owners.

This technical data may be subject to U.S. and international export, re-export, or transfer ("export") laws. Diversion contrary to U.S. and international law is strictly prohibited.

Qualcomm Technologies International, Ltd. (formerly known as Cambridge Silicon Radio Limited) is a company registered in England and Wales with a registered office at: Churchill House, Cambridge Business Park, Cowley Road, Cambridge, CB4 0WZ, United Kingdom.
Registered Number: 3665875 | VAT number: GB787433096

Revision history

Revision	Date	Description
1	April 2014	Initial release. Alternative document number CS-00314161-AN.
2	August 2014	Technical updates
3	June 2015	Updated for ADK 4.0
4	September 2016	Updated for ADK 4.1. Updated to conform to QTI standards.
5	April 2017	Updated demo License Key.
AF	October 2017	Added to the Content Management System. DRN updated to use Agile number. No change to technical content.
AG	February 2018	Added information about the aptX Low Latency codec ID change.

Contents

- Revision history 2
- 1 aptX overview 7
 - 1.1 aptX Classic vs aptX low latency vs aptX HD 7
- 2 Enabling aptX in the ADK Source application 8
 - 2.1 Enabling aptX Classic 8
 - 2.2 Enabling aptX Low Latency 10
 - 2.3 Enabling aptX HD 11
- 3 Enabling aptX in ADK sink applications 13
 - 3.1 aptX Classic 13
 - 3.2 aptX Low Latency 18
 - 3.3 aptX HD 22
- 4 Testing aptX 27
 - 4.1 Capabilities exchange 27
 - 4.2 aptX data over A2DP 29
 - 4.3 How to ensure aptX is running 29
 - 4.4 Verifying that the A2DP link is using aptX 30
- A AptX Low Latency (V2) Configuration 32
- Document references 34
- Terms and definitions 35

Tables

Table 3-1: aptX Low Latency DSP application variants..... 18

Table 4-1: Vendor and codec IDs.....27

Table A-1: AptX Low Latency V2 capabilities codec information..... 32

Figures

Figure 2-1: Enable the aptX codec in the source configuration tool.....	9
Figure 2-2: Sink makefile content to be uncommented.....	9
Figure 2-3: Configure the aptX Low Latency codec in the source configuration tool.....	10
Figure 2-4: Edit source.mak to copy the DSP aptX Low Latency DSP .kap file.....	11
Figure 2-5: Enable the aptX HD codec in the Source Configuration Tool.....	12
Figure 2-6: Sink makefile content to be uncommented.....	12
Figure 3-1: Insert project into workspace.....	14
Figure 3-2: xIDE Navigator after adding the aptX decoder DSP project.....	15
Figure 3-3: Enabling the aptX code in the VM from the xIDE Project Properties.....	16
Figure 3-4: Edit the makefile to copy the DSP aptX decoder .kap file.....	16
Figure 3-5: Configuring aptX as an A2DP optional codec using the Sink Configuration tool.....	17
Figure 3-6: Disable the back channel for aptX Low Latency A2DP only configurations.....	19
Figure 3-7: Navigation window with all three DSP Projects added to a VM application.....	19
Figure 3-8: Enable aptX Low Latency (aptX Sprint) in the xIDE Project Properties.....	20
Figure 3-9: Edit the makefile file to copy in the aptx_acl_sprint_decoder .kap file.....	20
Figure 3-10: Enabling aptX Low Latency in the Sink Configuration tool.....	21
Figure 3-11: 1 Insert project into workspace.....	22
Figure 3-12: xIDE Navigator after adding the aptX HD decoder DSP project.....	23
Figure 3-13: Enabling the aptX Code in the VM from the xIDE Project Properties.....	24
Figure 3-14: Edit Makefile to Copy the DSP aptX Decoder .kap File.....	24
Figure 3-15: Configuring aptX HD as an A2DP Optional Codec using the Sink Configuration tool.....	25
Figure 4-1: Air trace captured when aptX is enabled.....	28
Figure 4-2: Screenshot of the xIDE debugger when streaming with aptX Low Latency.....	29
Figure 4-3: Music Manager showing an aptX A2DP stream.....	30

Figure 4-4: Music Manager showing an aptX Low Latency A2DP stream.....	30
Figure 4-5: Music Manager showing an aptX HD A2DP stream.....	31

1 aptX overview

Qualcomm® aptX™ Classic and/or the aptX Low Latency and/or the aptX HD audio codec in the QTIL ADK can be used to support the development of aptX devices based on either CSR8675 or CSR8670 ICs.

1.1 aptX Classic vs aptX low latency vs aptX HD

The ADK includes three different aptX codecs:

1. aptX Classic: Which provides an end-to-end latency between 70 ms and 200 ms, depending on buffer settings in the source and sink devices.
2. aptX Low Latency: Which provides a fixed end-to-end latency of 40 ms \pm 3 ms.
3. aptX HD: Which provides a 24bit audio solution for improved sound quality compared with aptX Classic.

- NOTE**
- (1) aptX Low Latency needs to be enabled in both ends of the link to be selected by the application.
 - (2) In the rest of this document, the normal implementation of aptX is referred to as aptX Classic, and the new implementation for lower latency applications is referred to as aptX Low Latency. However, in the source code aptX Low Latency may also be referred to as aptX ACL Sprint.
 - (3) To ensure backward compatibility with other aptX devices on the market, aptX Classic is mandatory in any aptX enabled device. Do not develop a product that only supports aptX Low Latency. The newer 24-bit implementation is referred to as aptX HD.
 - (4) There is an updated version of aptX Low Latency codec (with a different codec ID). This version allows the source to configure the end-to-end latency. This version is not compatible with the previous aptX Low Latency codec (V1).
 - (5) For aptX HD, 24-bit audio is not available on the internal ADC/DAC. A digital interface must be used for 24-bit audio. Configuring the digital interfaces for 24-bit mode is covered in the ADK documentation.

2 Enabling aptX in the ADK Source application

How to enable the various aptX codecs in the ADK Source application is described in:

- [Enabling aptX Classic](#)
- [Enabling aptX Low Latency](#)
- [Enabling aptX HD](#)

NOTE To ensure backward compatibility, aptX Classic is mandatory in any aptX enabled device. Do not develop a product that only supports aptX Low Latency or aptX HD.

2.1 Enabling aptX Classic

To configure aptX classic in the Source application:

1. Install the aptX library

Although most of the code to enable aptX is provided by default in the ADK, the aptX library still requires an additional installer. Download the aptX installer from the ADK download page on [createpoint](#).

When downloaded, install the aptX add-on in the same folder as the ADK root folder. This adds the aptX libraries in the **<ADK>\kalimballib_sets\sdk\gordon** and **<ADK>\kalimballib_sets\sdk\rick** directories.

NOTE The libraries for aptX HD are only installed in the rick directory.

2. Enable aptX in the persistent store by configuring the Source as shown in Figure 2-1.

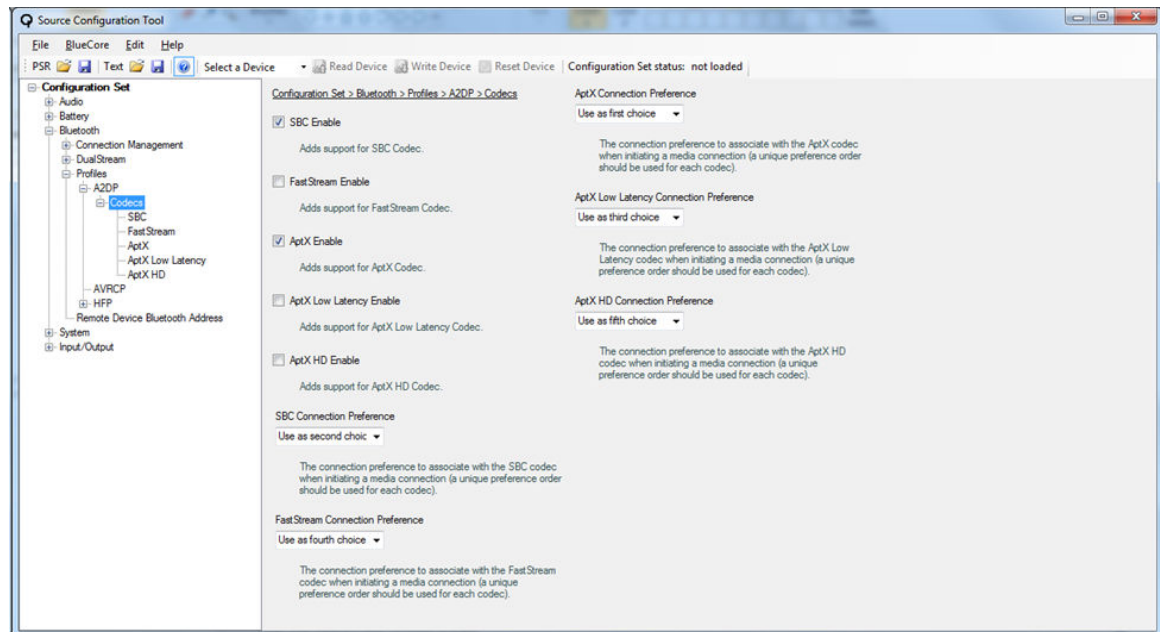


Figure 2-1 Enable the aptX codec in the source configuration tool

3. Add the aptX DSP project, `aptx_encoder.xip`, to the workspace using the **Insert Project into Workspace** option in the xIDE **Project** menu.

NOTE The aptX DSP project, `aptx_encoder.xip`, is located in the `kalimba\apps\la2dp_source` directory.

4. Edit the makefile.

To load the DSP image file within the IC memory, the makefile needs to copy the image file.

To do this, uncomment the appropriate lines in the `source.mak` file, see Figure 2-2.

```
## Include Classic aptX encoder - uncomment the following lines when enabling Classic aptX
image/aptx_encoder/aptx_encoder.kap :
    $(mkdir) image/aptx_encoder
    $(copyfile) ../../kalimba/apps/a2dp_source/image/aptx_encoder/aptx_encoder.kap $@

image.fs : image/aptx_encoder/aptx_encoder.kap
```

Figure 2-2 Sink makefile content to be uncommented

5. Build in xIDE by pressing **F5** on the keyboard.

NOTE This is the equivalent clicking **Build All** and then **Run** in the xIDE menu bar.

2.2 Enabling aptX Low Latency

To enable aptX Low Latency encoders in the ADK source application:

NOTE To ensure backward compatibility with other aptX devices on the market, aptX Classic is mandatory in any aptX enabled device. Do not develop a product with only aptX Low Latency.

1. Enable aptX Classic as described in [Enabling aptX Classic](#)
2. Configure aptX Low Latency in the persistent store by configuring the Source as shown in [Figure 2-3](#).

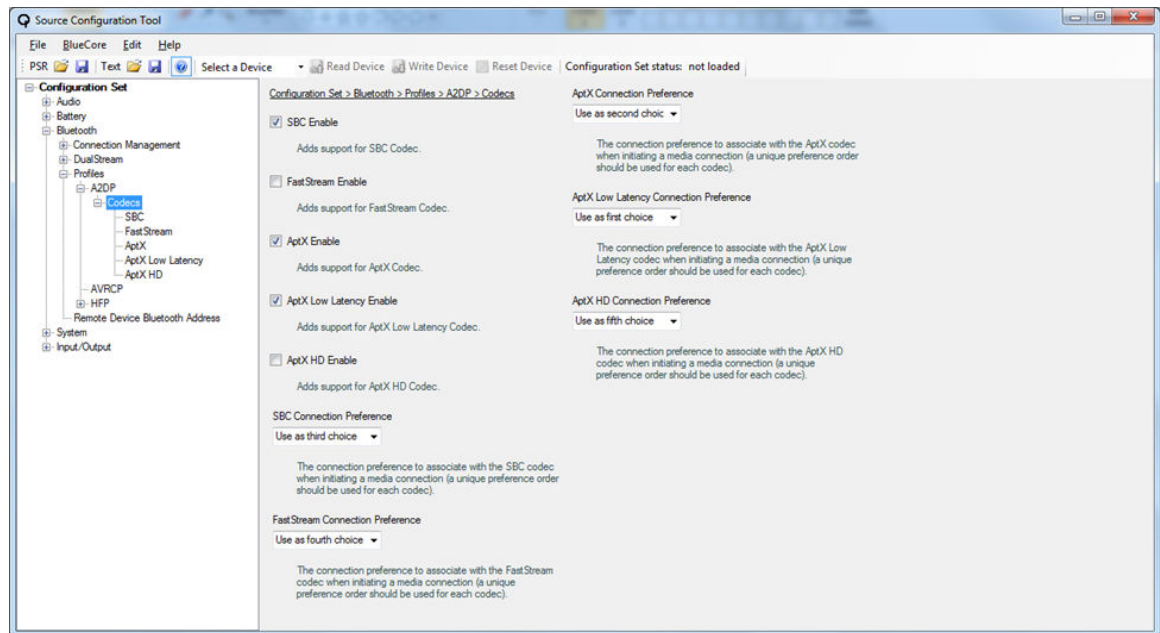


Figure 2-3 Configure the aptX Low Latency codec in the source configuration tool

3. Add the aptX DSP project, `aptx_acl_sprint_encoder.xip`, to your workspace using the **Insert Project into Workspace** option in the xIDE **Project** menu.

NOTE The aptX DSP project, `aptx_acl_sprint_encoder.xip`, is located in the `kalimbalapps\l2dp_source` directory.

4. Edit the makefile
 - a. To load the DSP image file within the IC memory, the makefile needs to copy the image file.
 - b. To do this, uncomment the appropriate lines in `source.mak`, see [Figure 2-4](#).

```
## Include Classic aptX encoder - uncomment the following lines when enabling Classic aptX
image/aptx_encoder/aptx_encoder.kap :
$(mkdir) image/aptx_encoder
$(copyfile) ../../kalimba/apps/a2dp_source/image/aptx_encoder/aptx_encoder.kap $$

image.fs : image/aptx_encoder/aptx_encoder.kap

## Include aptX Low Latency encoder - uncomment the following lines when enabling aptX Low Latency
image/aptx_acl_sprint_encoder/aptx_acl_sprint_encoder.kap :
$(mkdir) image/aptx_acl_sprint_encoder
$(copyfile) ../../kalimba/apps/a2dp_source/image/aptx_acl_sprint_encoder/aptx_acl_sprint_encoder.kap $$

image.fs : image/aptx_acl_sprint_encoder/aptx_acl_sprint_encoder.kap
```

Figure 2-4 Edit source.mak to copy the DSP aptX Low Latency DSP .kap file

5. Build in xIDE by pressing **F5** on the keyboard.

NOTE This is the equivalent clicking **Build All** and then **Run** in the xIDE menu bar.

Bidirectional feature with aptX Low Latency

An A2DP return channel from the sink to the source can be enabled with aptX Low Latency when available in both Source and Sink devices.

This feature is automatically enabled when the Source application is configured as a USB dongle.

2.3 Enabling aptX HD

To enable aptX HD encoders in the ADK source application:

NOTE To ensure backward compatibility with other aptX devices on the market, aptX Classic is mandatory in any aptX enabled device. Do not develop a product with only aptX HD.

1. Enable aptX Classic as described in [Enabling aptX Classic](#).
2. Enable the aptX HD codec in the Persistent Store by configuring the Source, see [Figure 2-5](#).

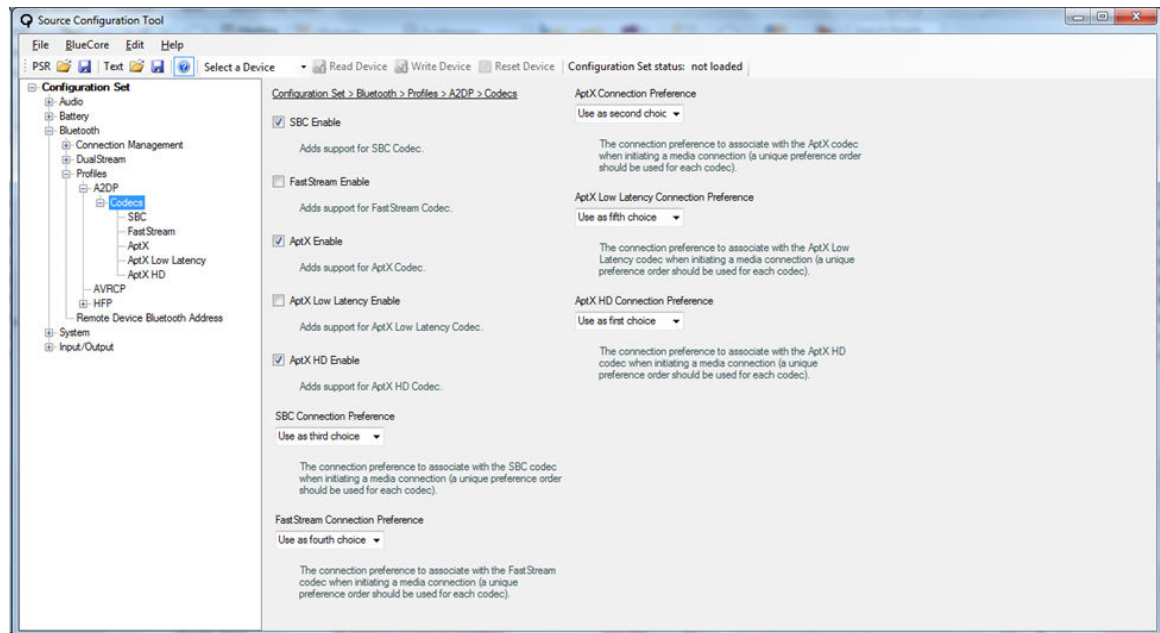


Figure 2-5 Enable the aptX HD codec in the Source Configuration Tool

3. Add the aptX DSP project, `aptxhd_encoder.xip`, to the workspace using the **Insert Project into Workspace** option in the xIDE **Project** menu.

NOTE The aptX DSP project, `aptxhd_encoder.xip`, is located in the `kalimba\apps\la2dp_source` directory.

4. Edit the makefile
 - a. To load the DSP image file within the IC memory, the makefile needs to copy the image file.
 - b. To do this, uncomment the appropriate lines in the `source.mak` file, see [Figure 2-6](#).

```
## Include aptX HD encoder - uncomment the following lines when enabling aptX HD
image/aptxhd_encoder/aptxhd_encoder.kap :
$(mkdir) image/aptxhd_encoder
$(copyfile) ../../kalimba/apps/a2dp_source/image/aptxhd_encoder/aptxhd_encoder.kap $@
image.fs : image/aptxhd_encoder/aptxhd_encoder.kap
```

Figure 2-6 Sink makefile content to be uncommented

5. Build in xIDE by pressing **F5** on the keyboard.

NOTE This is the equivalent clicking **Build All** and then **Run** in the xIDE menu bar.

3 Enabling aptX in ADK sink applications

How to enable the various aptX codecs in ADK sink applications, for example a headset, is described in:

- [aptX Classic](#)
- [aptX Low Latency](#)
- [aptX HD](#)

NOTE To ensure backward compatibility with other aptX devices on the market, aptX Classic is mandatory in any aptX enabled device. Do not develop a product with only aptX Low Latency or aptX HD.

3.1 aptX Classic

To configure aptX classic in a Sink application, for example a stereo headset:

1. Install the aptX library

Although most of the code to enable aptX is provided by default in the ADK, the aptX library still requires an additional installer. The aptX installer is available from the ADK download page on [createpoint](#).

When downloaded, install the aptX add-on in the same folder as your ADK root folder. This adds the aptX libraries in the `<ADK>\kalimba\lib_sets\sdk\gordon` and `<ADK>\kalimba\lib_sets\sdk\> rick` directories.

NOTE The libraries for aptX HD are only installed in the **rick** directory.

2. Add aptX Decoder DSP project:
 - a. Open the **Project** menu and select **Insert Project into Workspace**, as shown in Figure 3-1. Then locate and select the aptX project to be inserted.

NOTE The aptX DSP project, `aptx_decoder.xip` <ADK>/kalimbalapps/a2dp_sink

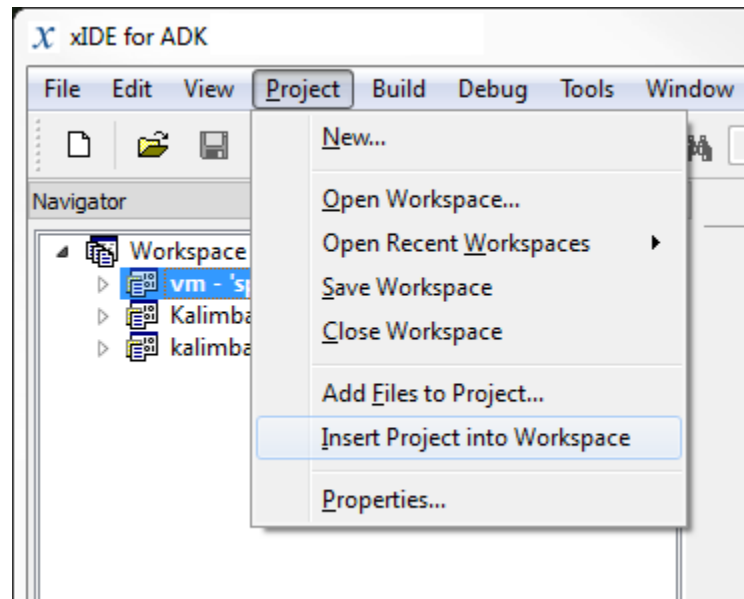


Figure 3-1 Insert project into workspace

The Qualcomm® Kalimba™ aptX decoder now appears in the xIDE **Navigator** window, see [Figure 3-2](#).

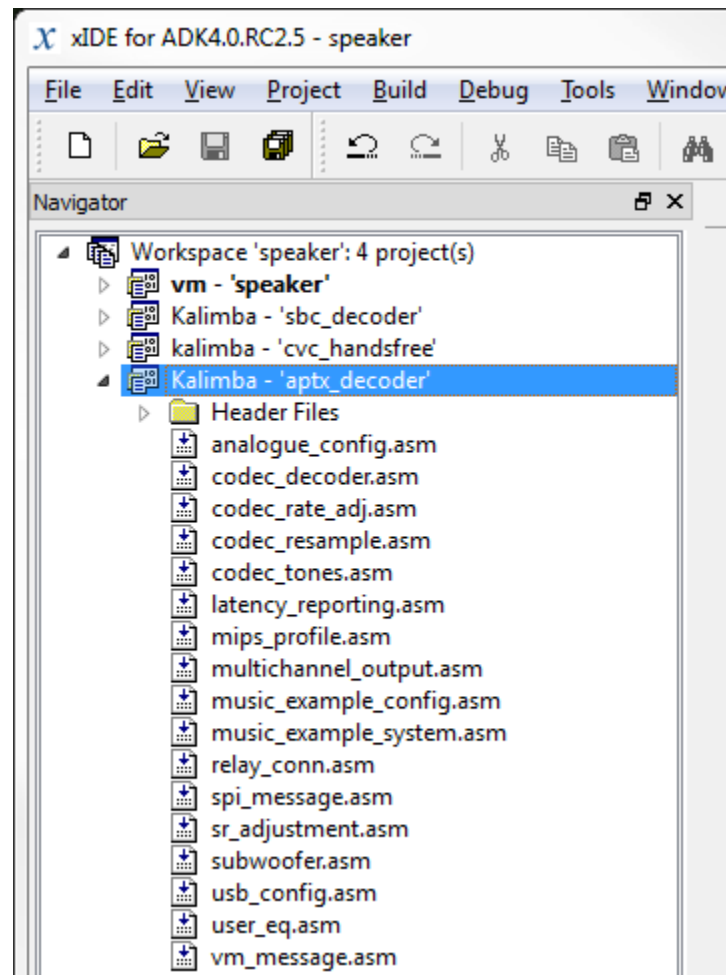


Figure 3-2 xIDE Navigator after adding the aptX decoder DSP project

3. Enable Extra Codec option.

The code that handles aptX is integrated by default in the ADK. However, the **Extra Codec** option in the **Project Properties**, needs to be enabled, see [Figure 3-3](#).

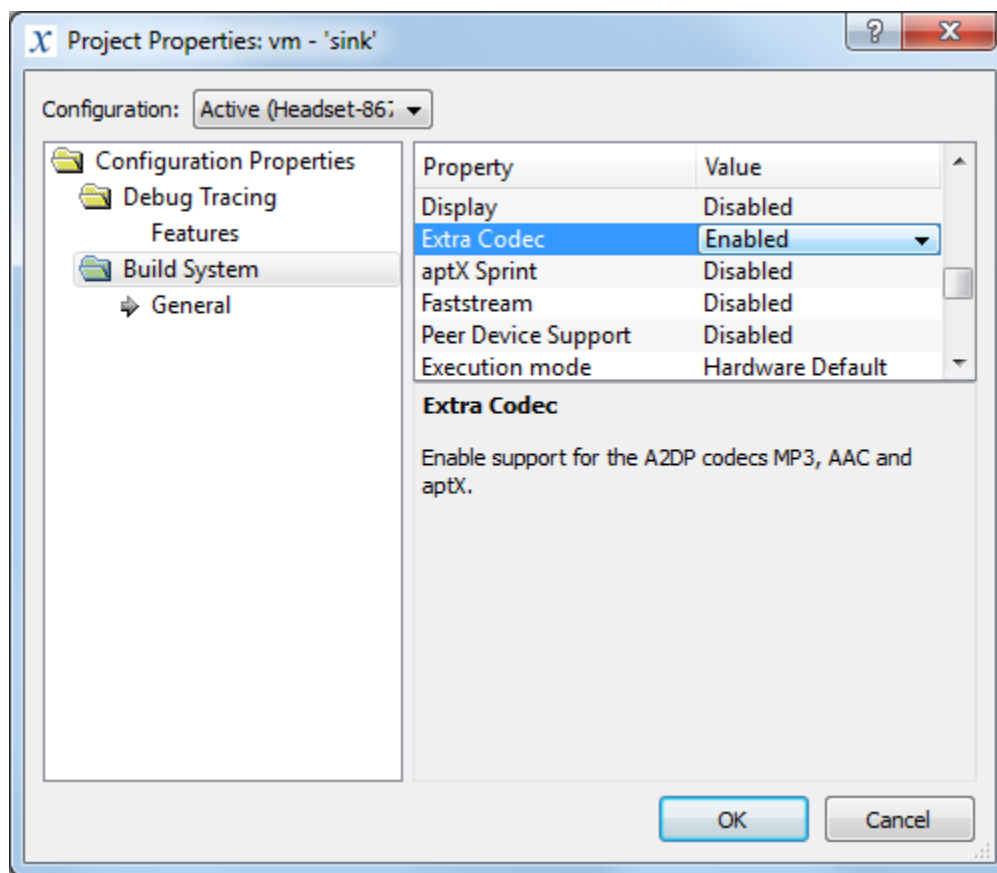


Figure 3-3 Enabling the aptX code in the VM from the xIDE Project Properties

4. Edit makefile.

To include the aptX decoder DSP binary into the IC image, the makefile needs to copy the .kap file from **<ADK>\kalimba\apps\aptx_decoder\image** to the **<ADK>\apps\sink\image** directory.

To do this, uncomment the appropriate line in the .mak file, see [Figure 3-4](#).

```

99
100 # copy in aptX decoder
101 image/ptx_decoder/ptx_decoder.kap :
102     $(mkdir) image/ptx_decoder
103     $(copyfile) ..\..\kalimba\apps\aptx_decoder\image\ptx_decoder\ptx_decoder.kap $@
104
105 image.fs : image/ptx_decoder/ptx_decoder.kap
106

```

Figure 3-4 Edit the makefile to copy the DSP aptX decoder .kap file

5. Build in xIDE by pressing **F5** on the keyboard.

NOTE This is the equivalent clicking **Build All** and then **Run** in the xIDE menu bar.

6. Configure the Persistent Store

Use the Sink Configuration Tool to enable the aptX Classic codec in the Sink application.

To do this:

- a. Navigate to the group:

Bluetooth > Profiles > A2DP > A2DP Optional Codecs Enabled

- b. Set the checkbox for aptX, see [Figure 3-5](#).

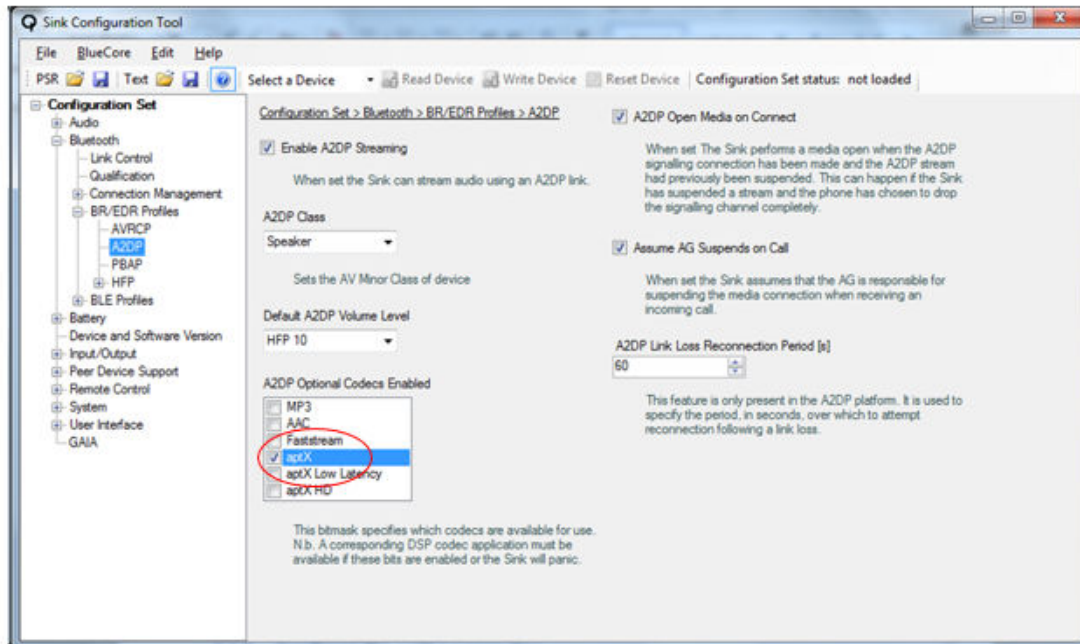


Figure 3-5 Configuring aptX as an A2DP optional codec using the Sink Configuration tool

7. Configure the aptX license key.

To unmute the audio output of the aptX decoder library, a license key is required in PSKEY_DSP20.

For evaluation, use the license key: 3465 A979 BC24 823B 4BA1 and set the Bluetooth address of the device in the range 0002 5b 00ff01 to 0002 5b 00ff05.

NOTE Contact QTIL to obtain a license key that can be used in mass production.

There are two license key options for aptX:

1. To enable both aptX Classic, aptX HD and aptX Low Latency.
2. To enable aptX Classic and aptX HD only.

The evaluation key provided above for evaluation purposes enables both implementations.

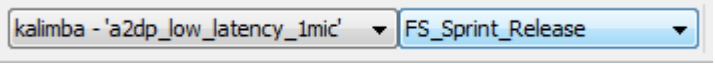
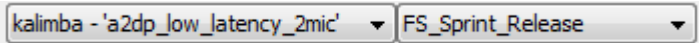
NOTE There is no need for a license key when using the aptX encoder library.

3.2 aptX Low Latency

To configure aptX Low Latency in a Sink application, for example a stereo headset.

1. Add an aptX Low Latency decoder DSP project by selecting the DSP project to add to the workspace, see [Table 3-1](#).

Table 3-1 aptX Low Latency DSP application variants

Decoder	Description
<code>aptx_acl_sprint_decoder</code>	aptX Low Latency decoder without a back channel. Used for simple tests with A2DP streaming only.
<code>a2dp_low_latency_1mic</code>	Headset with back channel, 1-mic configuration This project must be configured for either FS (FastStream) only or FS and Sprint (aptX Low Latency), if it is included in the workspace. This can be configured using the drop-down on the xIDE toolbar: 
<code>a2dp_low_latency_2mic</code>	Headset with back channel, 2-mic configuration This project must be configured for either FS (FastStream) only or FS and Sprint (aptX Low Latency), if it is included in the workspace. This can be configured using the drop-down on the xIDE toolbar:  <p>NOTE If you intend to use a 2mic low latency A2DP application, you need to use the CSR8675 device. The CSR 8670 is not compatible with this application.</p>

NOTE An A2DP return channel from the sink to the source can be enabled with aptX Low Latency when available in both source and sink devices. The Sink part is configured by using the back channel capable DSP projects.

The back channel capable DSP projects must be configured if they are used in the project. They are added by default in the Stereo Headset example application provided in the ADK. To build a

default headset project without using a back channel ensure the Sink Configuration Tool is used to disable the back channel, as shown in Figure 3-6.

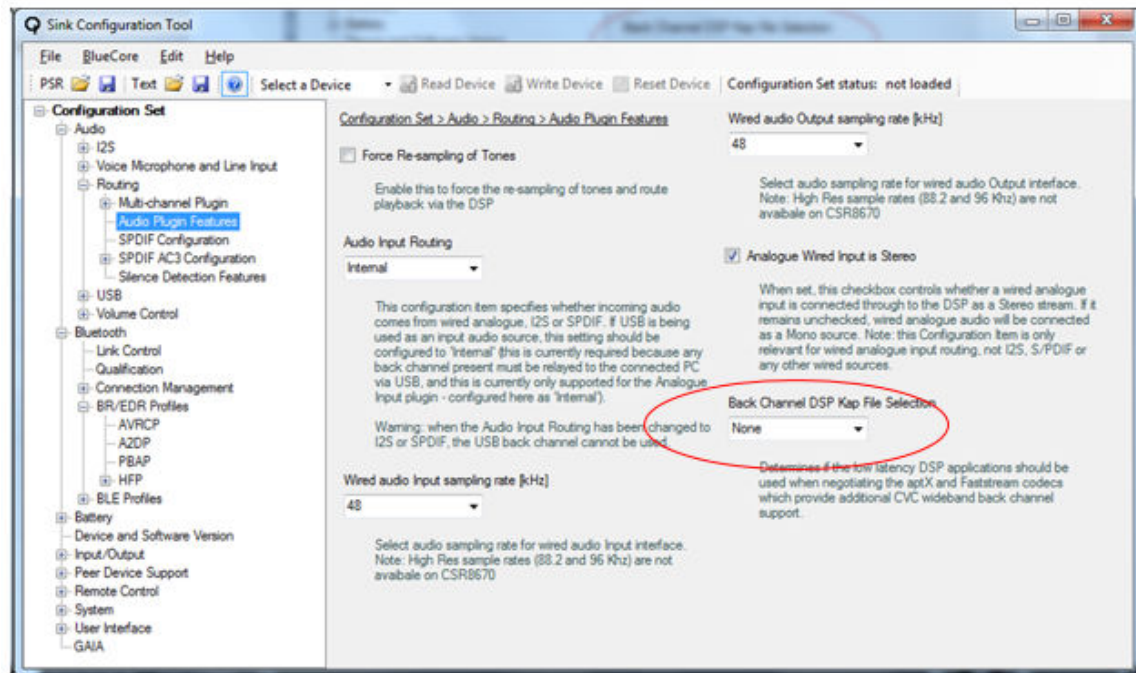


Figure 3-6 Disable the back channel for aptX Low Latency A2DP only configurations

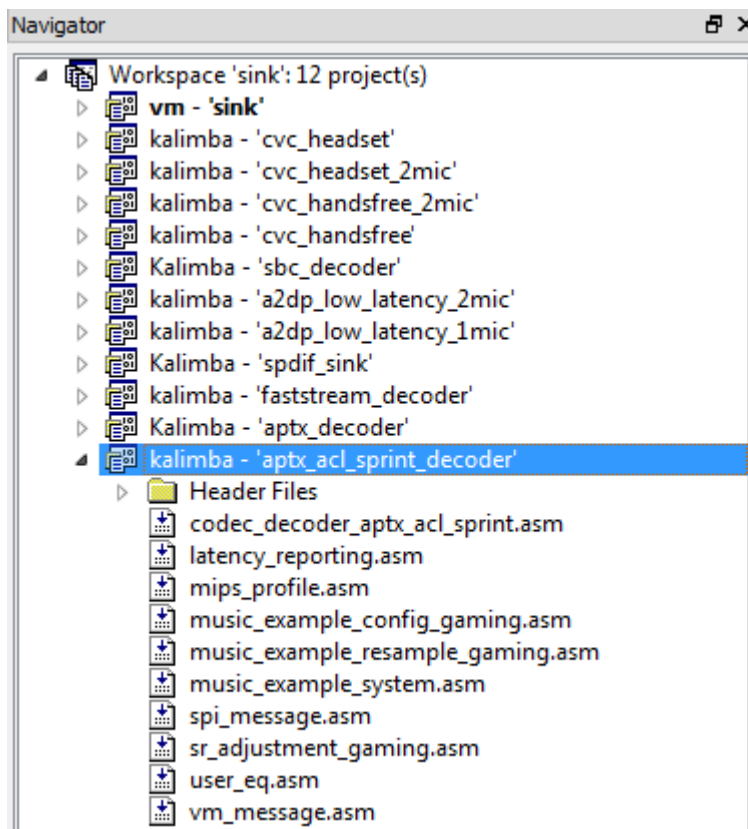


Figure 3-7 Navigation window with all three DSP Projects added to a VM application

2. Enable aptX sprint option.

The code that handles aptX is integrated by default in the ADK. However, the **aptX Sprint** option in the **Project Properties**, needs to be enabled, see [Figure 3-8](#).

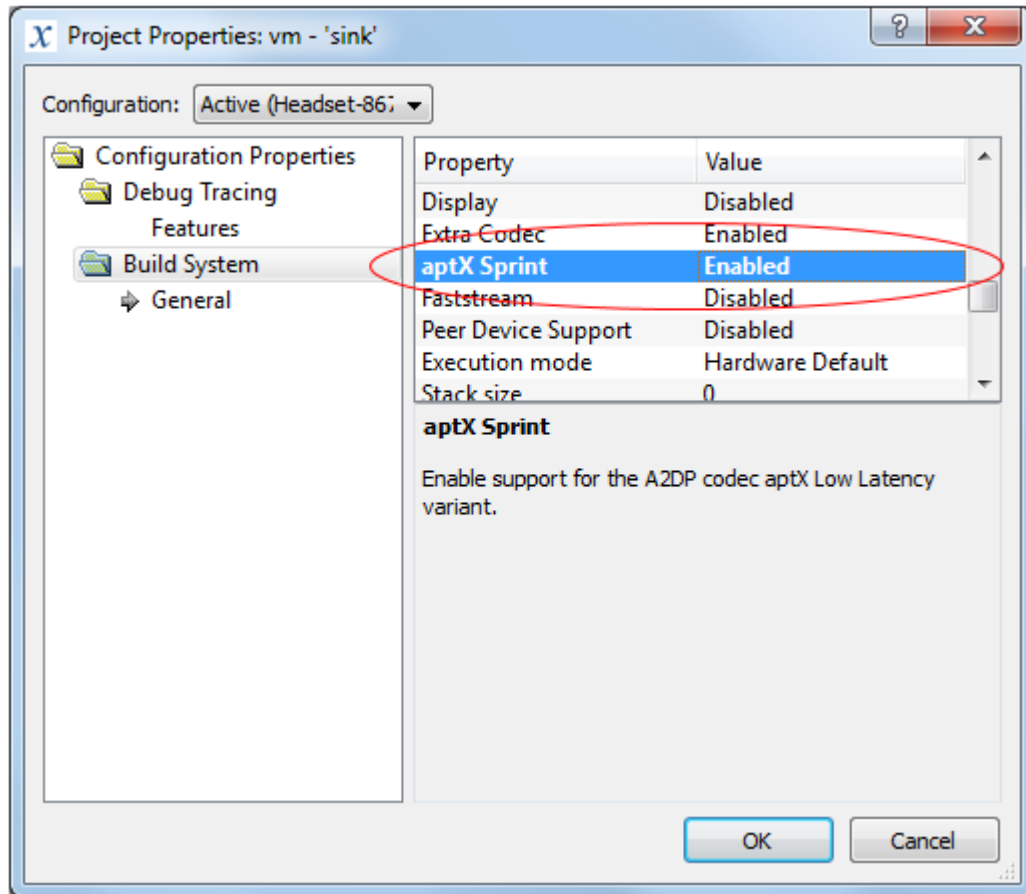


Figure 3-8 Enable aptX Low Latency (aptX Sprint) in the xIDE Project Properties

3. Edit the sink makefile.

To include the aptX decoder DSP binary into the IC image, the makefile needs to copy the .kap file from <ADK>\kalimba\apps\l2dp_sink\image to the <ADK>\apps\sink\image directory.

To do this, uncomment the appropriate line in the .mak file, see [Figure 3-9](#).

```

107# copy in aptx_acl_sprint_decoder
108# Note: If using aptX Low Latency you should also include the aptx_decoder.kap above as well
109image/ptx_acl_sprint_decoder/ptx_acl_sprint_decoder.kap :
110    $(mkdir) image/ptx_acl_sprint_decoder
111    $(copyfile) ../../kalimba/apps/l2dp_sink/image/ptx_acl_sprint_decoder/ptx_acl_sprint_decoder.kap $@
112
113image.fs : image/ptx_acl_sprint_decoder/ptx_acl_sprint_decoder.kap
114

```

Figure 3-9 Edit the makefile file to copy in the aptx_acl_sprint_decoder .kap file

4. Configure the Persistent Store

Use the Sink Configuration Tool to enable the aptX Low Latency codec in the Sink application.

To do this:

- a. Navigate to the group:
Bluetooth > Profiles > A2DP > A2DP Optional Codecs Enabled
- b. Set the checkboxes for aptX (mandatory) and aptX Low Latency, see [Figure 3-10](#).

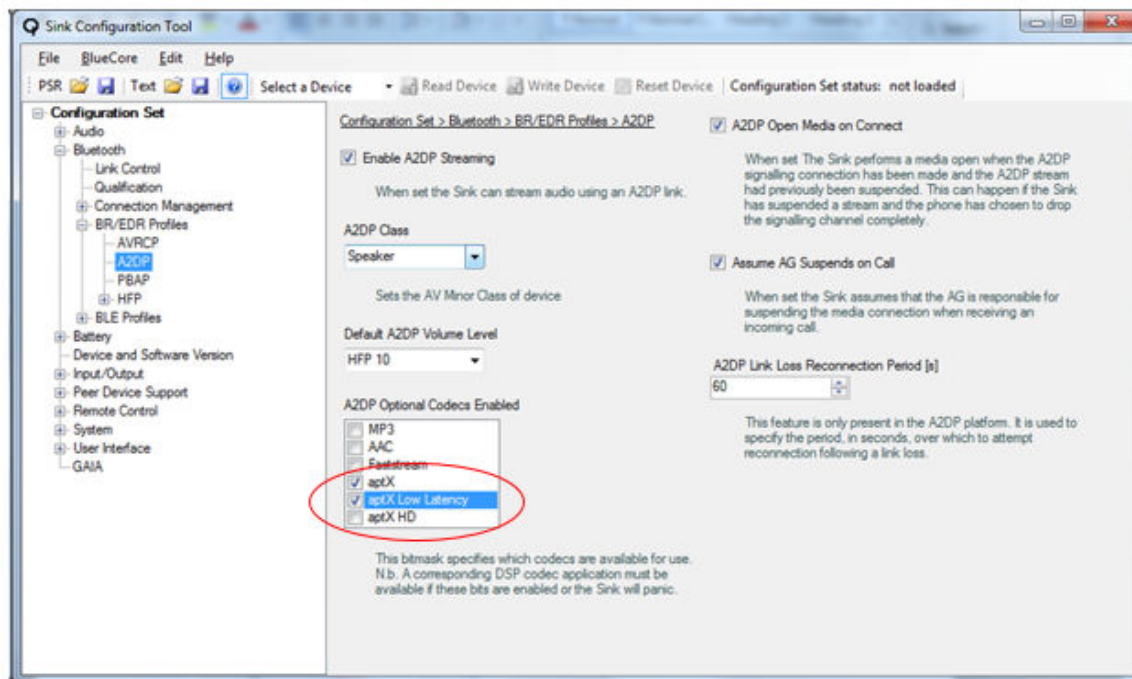


Figure 3-10 Enabling aptX Low Latency in the Sink Configuration tool.

5. Build in xIDE by pressing **F5** on the keyboard.

NOTE This is the equivalent clicking **Build All** and then **Run** in the xIDE menu bar.

6. Configure the aptX license key.

To unmute the audio output of the aptX decoder library, a license key is required in PSKEY_DSP20.

For evaluation, use the license key: 3465 A979 BC24 823B 4BA1 and set the Bluetooth address of the device in the range 0002 5b 00ff01 to 0002 5b 00ff05.

NOTE Contact QTIL to obtain a license key that can be used in mass production.

There are two license key options for aptX:

1. To enable both aptX Classic, aptX HD and aptX Low Latency.
2. To enable aptX Classic and aptX HD only.

The evaluation key provided above for evaluation purposes enables both implementations.

When using `a2dp_low_latency_1mic` or `a2dp_low_latency_2mic`, a valid cVc license key is also required in PSKEY_DSP48.

3.3 aptX HD

To configure aptX HD in a Sink application, for example a stereo headset:

1. Install the aptX HD library

Although most of the code to enable aptX is provided by default in the ADK, the aptX library still requires an additional installer. The aptX installer is available from the ADK download page on [createpoint](#).

When downloaded, install the aptX add-on in the same folder as your ADK root folder. This adds the aptX libraries in the **<ADK>\kalimba\lib_sets\sdk\gordon** and **<ADK>\kalimba\lib_sets\sdk\There is no need for a license key when using the aptX encoder library. > rick** directories.

2. Add aptX HD decoder DSP project

To do this, open the **Project** menu and select **Insert Project into Workspace**, as shown in [Figure 3-11](#). Then locate and select the aptX project to be inserted.

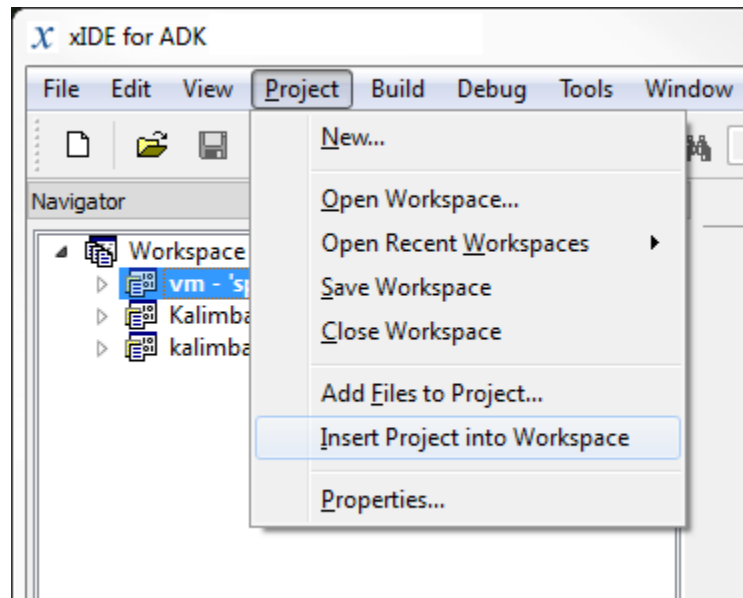


Figure 3-11 1 Insert project into workspace

NOTE There is no need for a license key when using the aptX encoderThe aptX HD decoder DSP project, `aptxhd_decoder.xip`, is located in the **<ADK>/kalimba\apps\la2dp_sink** directory.

The Kalimba aptX HD decoder now appears in the xIDE **Navigator** window, see [Figure 3-12](#).

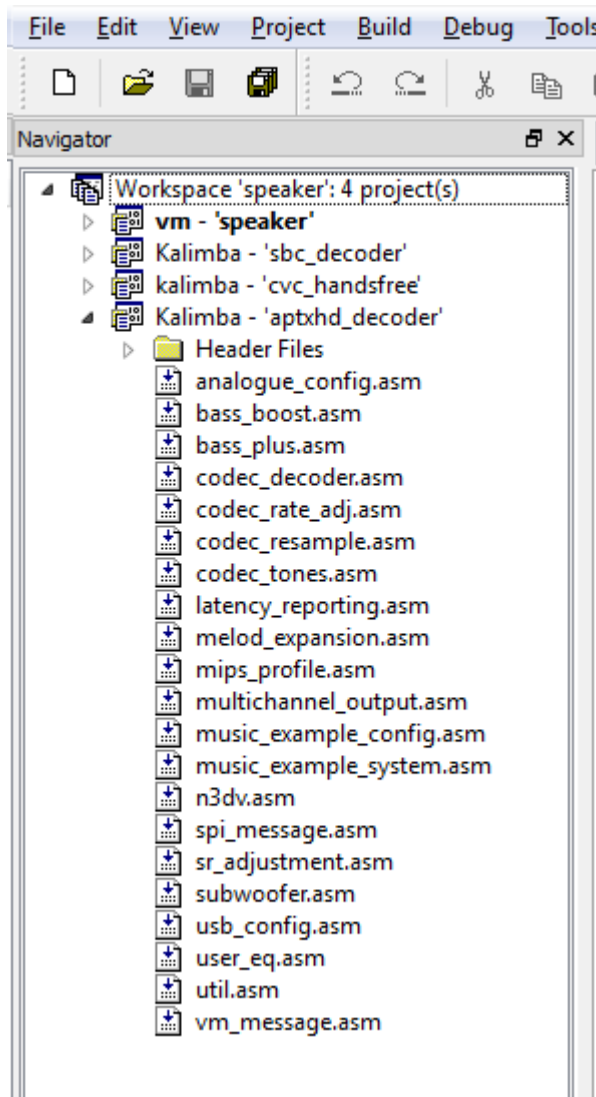


Figure 3-12 xIDE Navigator after adding the aptX HD decoder DSP project

3. Enable Extra Codec option

The code that handles aptX HD is integrated by default in the ADK. However, the **Extra Codec** option in the **Project Properties**, needs to be enabled, see [Figure 3-13](#).

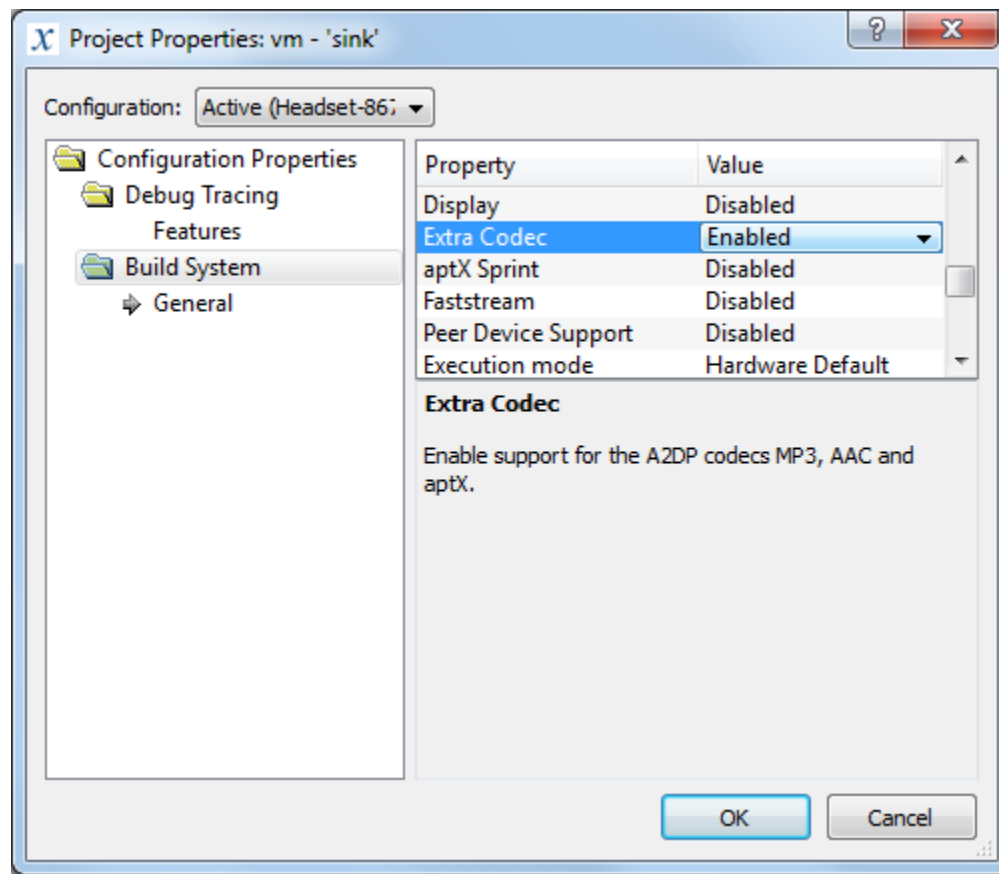


Figure 3-13 Enabling the aptX Code in the VM from the xIDE Project Properties

4. Edit makefile

To include the aptX decoder DSP binary into the IC image, the makefile needs to copy the .kap file from <ADK>\kalimba\apps\aptxhd_decoder\image to the <ADK>\apps\sink\image directory.

To do this, uncomment the appropriate line in the .mak file, see [Figure 3-14](#).

```
# copy in aptX HD decoder
image/ptxhd_decoder/ptxhd_decoder.kap :
$(mkdir) image/ptxhd_decoder
$(copyfile) ../../kalimba/apps/a2dp_sink/image/ptxhd_decoder/ptxhd_decoder.kap $@

image.fs : image/ptxhd_decoder/ptxhd_decoder.kap
```

Figure 3-14 Edit Makefile to Copy the DSP aptX Decoder .kap File

5. Build in xIDE by pressing **F5** on the keyboard.

NOTE This is the equivalent clicking **Build All** and then **Run** in the xIDE menu bar.

6. Configure the Persistent Store using the Sink Configuration Tool to enable the aptX HD codec in the Sink application.

To do this:

- a. Navigate to the group:

Bluetooth > Profiles > A2DP > A2DP Optional Codecs Enabled

- b. Set the checkbox for aptX and aptX HD, see [Figure 3-15](#).

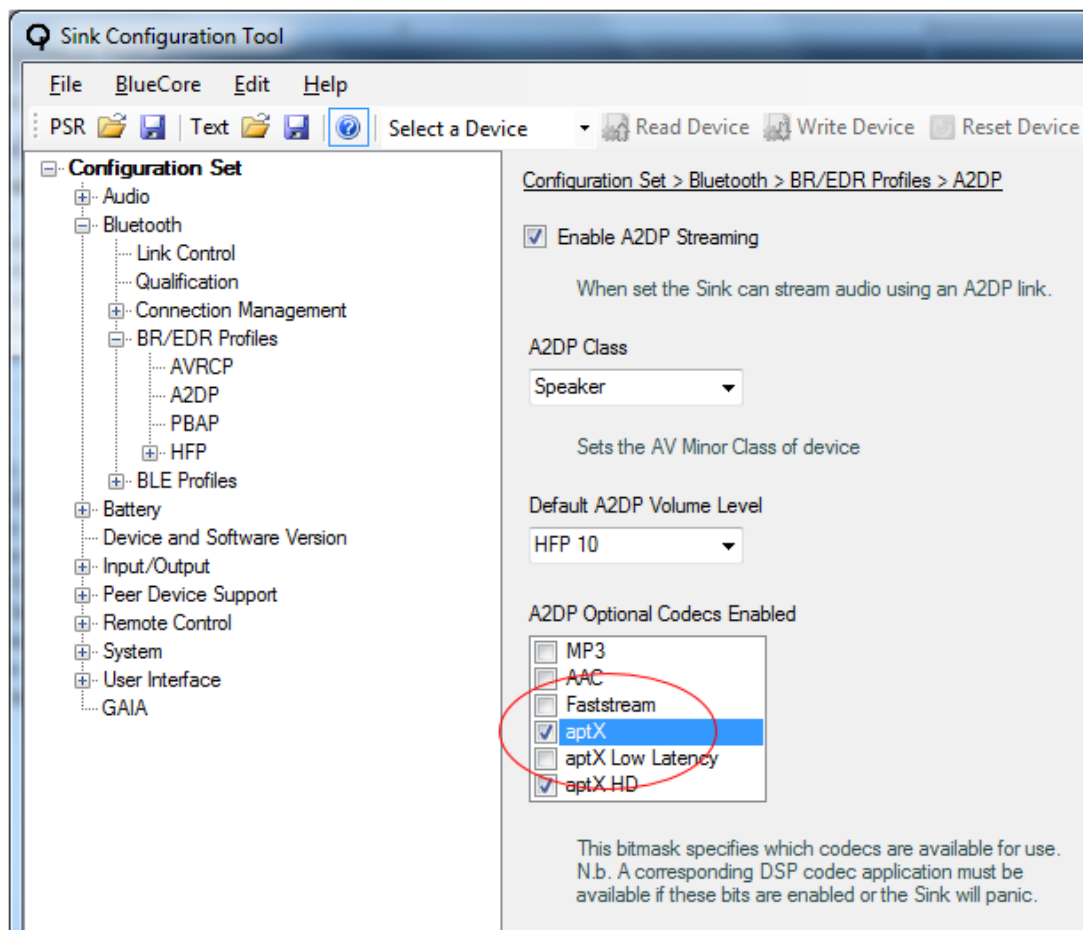


Figure 3-15 Configuring aptX HD as an A2DP Optional Codec using the Sink Configuration tool

7. Configure the aptX license key.

To un-mute the audio output of the aptX HD decoder library, a license key is required in PSKEY_DSP20.

For evaluation, use the license key: 3465 A979 BC24 823B 4BA1 and set the Bluetooth address of the device in the range 0002 5b 00ff01 to 0002 5b 00ff05.

NOTE Contact QUIL to obtain a license key that can be used in mass production.

There are two license key options for aptX:

- To enable both aptX Classic, aptX HD and aptX Low Latency.
- To enable aptX Classic and aptX HD only.

NOTE The evaluation key provided above for evaluation purposes enables all implementations.

4 Testing aptX

There are a number of ways to confirm that aptX is being used on an A2DP link, see:

- [Capabilities exchange](#)
- [aptX data over A2DP](#)
- [How to ensure aptX is running](#)
- [Verifying that the A2DP link is using aptX](#)

4.1 Capabilities exchange

The aptX codec is negotiated during connection time through the capabilities exchange process in AVDTP signaling. See the *Audio/Video Distribution Transport Protocol Specification v1.4* and *Advanced Audio Distribution Profile Specification v1.2* for more details.

Each Vendor specific A2DP codec in A2DP uses a unique Codec and Vendor ID. See Section 4.7 in the *Advanced Audio Distribution Profile Specification v1.2* for more information.

[Table 4-1](#) lists the IDs used by the aptX codecs.

Table 4-1 Vendor and codec IDs

Variant	Vendor ID	Codec ID
aptX Classic	0x4f 00 00 00	0x01 00
aptX Low latency (V1)	0x0a 00 00 00	0x02 00
aptX Low latency (V2)	0xd7 00 00 00	0x02 00

NOTE An extra codec ID for aptX Low latency has been added from ADK 4.2 onwards. There are now two variants of the codec ID, used to differentiate between the versions of the aptX Low Latency. The additional variant of the codec allows the source to configure the desired latency for the link. Both codecs are offered for ADK 4.2 based sinks. However, only the V2 aptX Low Latency codec is supported in the ADK 4.2 source. It will not select aptX Low Latency when negotiating with a sink that only offers V1 aptX Low Latency (i.e. one based on versions of ADK prior to 4.2). It will fall back to aptX Classic instead. The aptX V2 capabilities allow the source to configure aspects of the sink latency. For further details, see [AptX Low Latency \(V2\) Configuration](#). Qualcomm recommends that no source should offer both V1 and V2 of aptX Low Latency. Each source should only offer one or the other codec variant. It is recommended that, where available (i.e. from ADK 4.2 onwards), the V2 codec should be used by the source application.

Figure 4-1 shows a screenshot of a frontline FTS4BT Air Trace when aptX Classic is enabled in both Source and Sink devices:



Figure 4-1 Air trace captured when aptX is enabled

In the GET_CAPABILITIES commands, the Slave device enumerates the details of each codec it supports. The image on the left shows the details of the GET_CAPABILITIES response including aptX IDs, frame #146.

Then, in frame #147 on the right of the image, the Master device sends a SET_CONFIGURATION command to set the connection using one of the codecs enumerated by the slave.

At that stage, priority is given to the codecs as follows:

1. aptX Low Latency, if both devices are compatible with aptX Low Latency (same version).

NOTE The source should only be compatible with one of the codec variants of aptX Low Latency. If the sink is of an ADK version prior to ADK 4.2 (4.1 and below), then V2 codec variant will not be available and the negotiation will fall back to aptX Classic.

2. aptX Classic, if both devices are compatible with aptX Classic and one or both of the devices is not compatible with aptX Low Latency.
3. Any other Vendor-specific A2DP codec, optional codec, or the default SBC codec.

NOTE Both Source and Sink device can initiate the capabilities exchange. It is important to ensure that a device uses aptX whenever it initiates the connection, but it must also enumerate aptX when the other device is the initiator.

4.2 aptX data over A2DP

When the connection is initiated, the Source device can start sending the aptX codewords over the air. The bitrate of aptX (non-HD) is fixed at 352.8 kbps@44.1 KHz FS and 384 kbps@48 KHz FS. The bitrate of aptX HD is fixed at 529.2 kbps@44.1 KHz FS and 576 kbps@48 KHz FS. Unlike SBC, the aptX codec is frameless and therefore does not have any frame structure.

4.3 How to ensure aptX is running

Check aptX is running in the air trace

Check aptX is selected in the SET_CONFIGURATION command.

Checking aptX is running using xIDE debugger

When a connection has been initiated and streaming has started, the headset loads the DSP image corresponding to the selected codec. To check which DSP decoder has been loaded in the DSP memory select **Debug Windows** in the **View** menu and view the **Registers** window, see Figure 4-2.

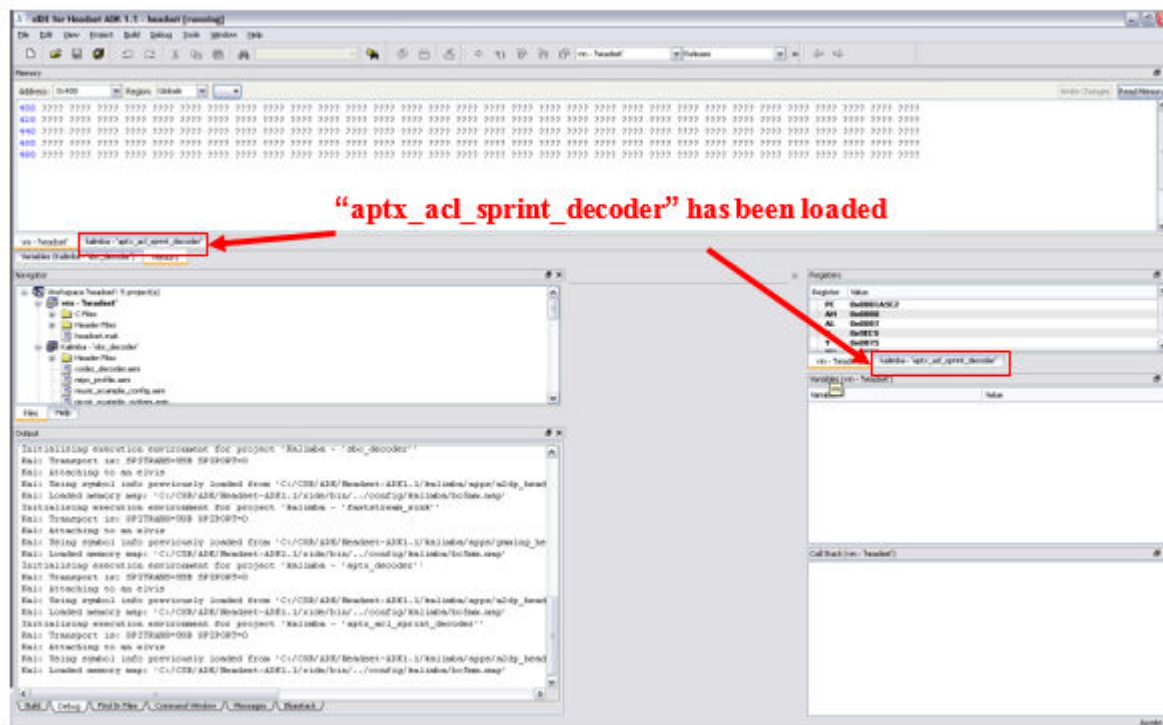


Figure 4-2 Screenshot of the xIDE debugger when streaming with aptX Low Latency

4.4 Verifying that the A2DP link is using aptX

The version of aptX being used can be checked using the Music Manager in the Universal Front End tool. Figure 4-3 shows a screenshot of the Music Manager when aptX Classic is running.

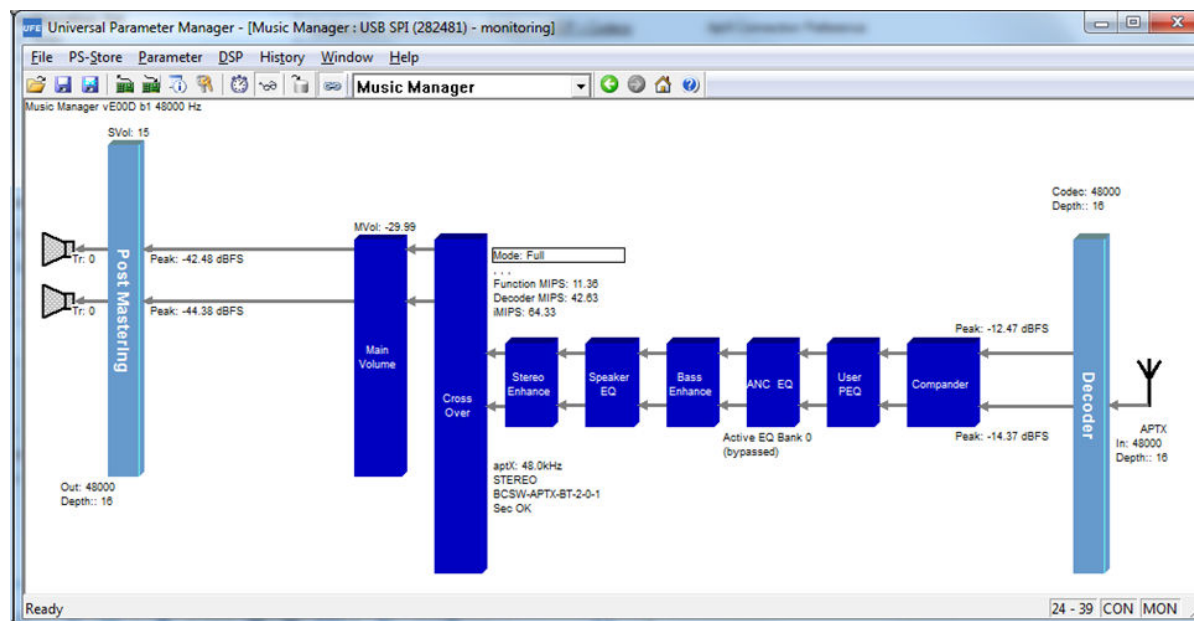


Figure 4-3 Music Manager showing an aptX A2DP stream

Figure 4-4 shows a screenshot of the Music Manager when aptX Low Latency is running.

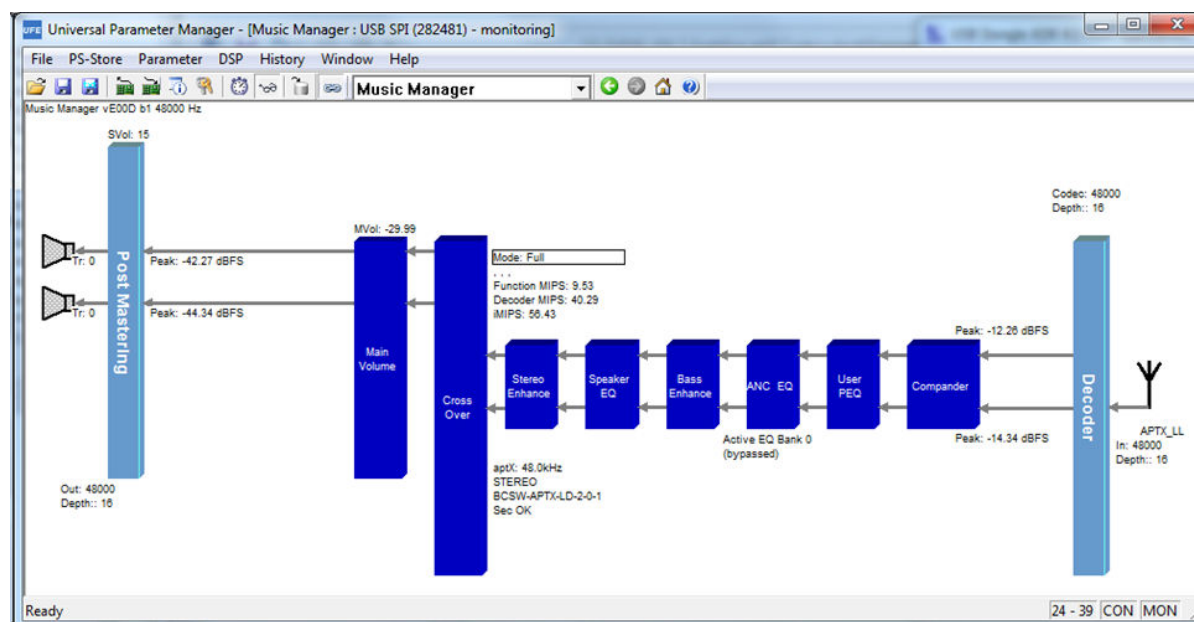


Figure 4-4 Music Manager showing an aptX Low Latency A2DP stream

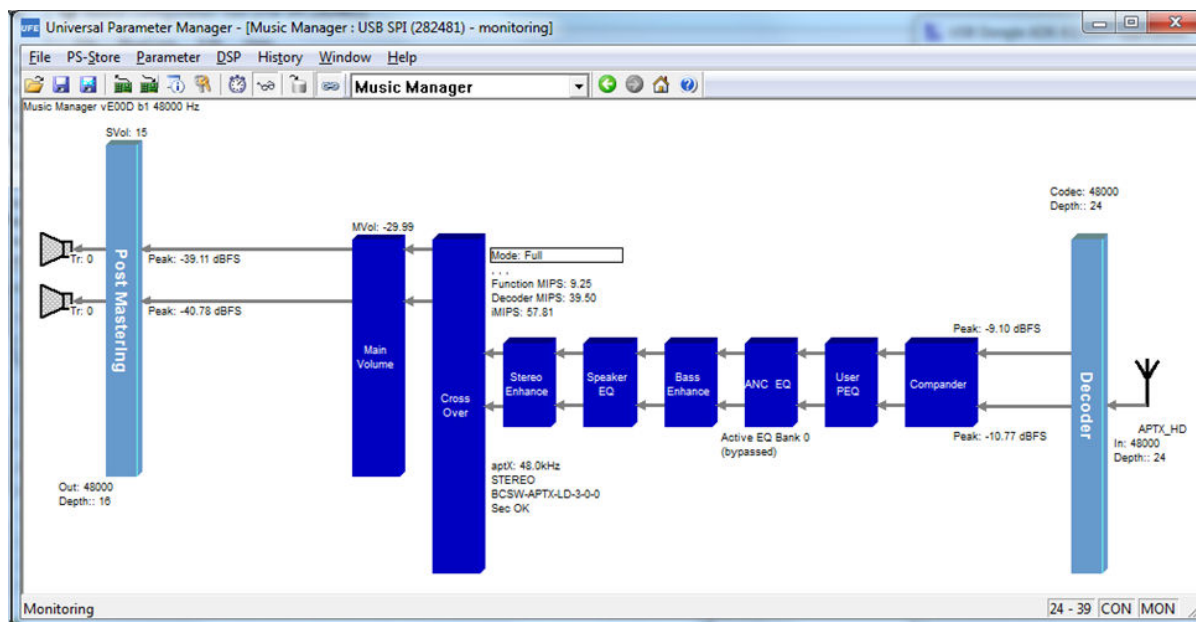


Figure 4-5 Music Manager showing an aptX HD A2DP stream

A AptX Low Latency (V2) Configuration

The aptX Low Latency (V2) capabilities exchange now include parameters that affect the latency of the link. [Table A-1](#) lists the structure of the AVDTP capabilities for the AptX Low Latency V2 codec.

Table A-1 AptX Low Latency V2 capabilities codec information

AVDTP capabilities (AptX Low Latency V2)	Values
QTIL Vendor ID 0	0xD7
QTIL Vendor ID 1	0x00
QTIL Vendor ID 2	0x00
QTIL Vendor ID 3	0x00
aptX Low Latency Codec ID 0	0x02
aptX Low Latency Codec ID 1	0x00
aptX Sampling Frequency Stereo Mode ^a	0x12
aptX Low Latency Voice New Caps	0x03
RESERVED	0x00
Target Codec Level LSB	0xB4
Target Codec Level MSB	0x00
Initial Codec Level LSB	0x7C
Initial Codec Level MSB	0x01
Maximum Rate	0x32
Average Period	0x01
Good Working Buffer Level LSB	0xB4
Good Working Buffer Level MSB	0x00

^a This value will be 0x22 for a sampling rate of 44.1 kHz, with stereo audio. The 0x12 represents a 48 kHz stereo signal.

The sink uses the following elements of the new capabilities:

- Target codec level (code words): Targeted average level. This will significantly affect the overall latency. A larger value gives a bigger latency. Smaller gives a lower latency level. Note that this will also impact on the robustness of the link.
- Averaging period: This denotes the period (in seconds) over which the buffer level is averaged.

- **Maximum rate (in increments of 50 ppm):** This is the maximum warping rate that the sink can use.
- **Good working buffer (words):** Lowest average level, below which poor link detection will be triggered. This is defined in `source_a2dp.c` as:

```

/* APT-X Low Latency Stream-End Point Capabilities */
static const uint8 a2dp_aptxLowLatency_caps_source[] = {
    AVDTP_SERVICE_MEDIA_TRANSPORT,
    0,
    AVDTP_SERVICE_MEDIA_CODEC,
    19,
    AVDTP_MEDIA_TYPE_AUDIO<<2,
    AVDTP_MEDIA_CODEC_NONA2DP,
    A2DP_QTI_VENDOR_ID0,
    A2DP_QTI_VENDOR_ID1,
    A2DP_QTI_VENDOR_ID2,
    A2DP_QTI_VENDOR_ID3,
    A2DP_APTX_LOWLATENCY_CODEC_ID0,
    A2DP_APTX_LOWLATENCY_CODEC_ID1,
    #if (defined ANALOGUE_INPUT_DEVICE && defined BC5_MULTIMEDIA)
    A2DP_APTX_SAMPLING_FREQ_44100 | A2DP_APTX_CHANNEL_MODE_STEREO,
    #else
    A2DP_APTX_SAMPLING_FREQ_48000 | A2DP_APTX_CHANNEL_MODE_STEREO,
    #endif
    A2DP_APTX_LOWLATENCY_VOICE_16000 | A2DP_APTX_LOWLATENCY_NEW_CAPS,
    A2DP_APTX_LOWLATENCY_RESERVED,
    A2DP_APTX_LOWLATENCY_TCL_LSB,
    A2DP_APTX_LOWLATENCY_TCL_MSB,
    A2DP_APTX_LOWLATENCY_ICL_LSB,
    A2DP_APTX_LOWLATENCY_ICL_MSB,
    A2DP_APTX_LOWLATENCY_MAX_RATE,
    A2DP_APTX_LOWLATENCY_AVG_TIME,
    A2DP_APTX_LOWLATENCY_GWBL_LSB,
    A2DP_APTX_LOWLATENCY_GWBL_MSB,
    AVDTP_SERVICE_CONTENT_PROTECTION,
    2,
    AVDTP_CP_TYPE_SCMS_LSB,
    AVDTP_CP_TYPE_SCMS_MSB
};

```

Document references

Document	Reference
<i>ADK Audio Sink User Guide</i>	80-CF316-1/CS-00236868-UG
<i>ADK Configuration Tool User Guide</i>	80-CT554-1/CS-00401879-UG
<i>ADK 4.3 Audio Source Application User Guide</i>	80-CF409-1/CS-00406741-UG
<i>ADK 4.3 Source Configuration Tool User Guide</i>	80-CF419-1/CS-00406798-UG
<i>Audio/Video Distribution Transport Protocol Specification v1.5</i>	www.bluetooth.org
<i>Advanced Audio Distribution Profile Specification v1.3</i>	www.bluetooth.org

Terms and definitions

Term	Definition
A2DP	Advanced Audio Distribution Profile
ADK	Audio or Application Development Kit
aptX	An A2DP return channel from the sink to the source can be enabled with aptX Low Latency when available in both source and sink devices. The Sink part is configured by using the back channel capable DSP projects.
AVDTP	Audio/Video Distribution Transport Protocol
BlueCore	Group term for the range of QTIL Bluetooth wireless technology ICs
Bluetooth	Set of technologies providing audio and data transfer over short-range radio connections
DSP	Digital Signal Processor
FS	FastStream
IC	Integrated Circuit
ID	IDentifier
PS	Persistent Store
QTIL	Qualcomm Technologies International, Ltd.
USB	Universal Serial Bus
VM	Virtual Machine
xIDE	The QTIL Integrated Development Environment