

bluetooth_mgr

- [Basic Bluetooth Architecture in RDK](#)
 - [Sequence of Steps](#)
- [btrMgrStreamOutCoreTest](#)
- [API Documentation](#)

Bluetooth Provides an interface to port any Bluetooth application on RDK. Bluetooth Manager (An RDK component) interfaces with BlueZ through the D-Bus API, so there is no direct linking of the BlueZ library with Bluetooth Manager.

In RDK, Bluetooth is comprised of a Core module (BTCore) and a Manager module (BTMgr). The BTCore module controls basic Bluetooth functionality such as scanning, pairing, and connecting. The BTMgr module uses the services of the BTCore module for performing audio streaming. RDK has two utilities for testing. A btrCoreTest utility tests the BTCore functions. A btrMgrStreamOutCoreTest utility tests BTMgr functions. Both utilities share some common functionality. For example, it is possible to perform a device scan using either utility.

Bluez

In its current implementation, the RDK Bluetooth uses the Bluez stack. It is possible that another stack (e.g. Broadcom's BSA) could be used in the future.

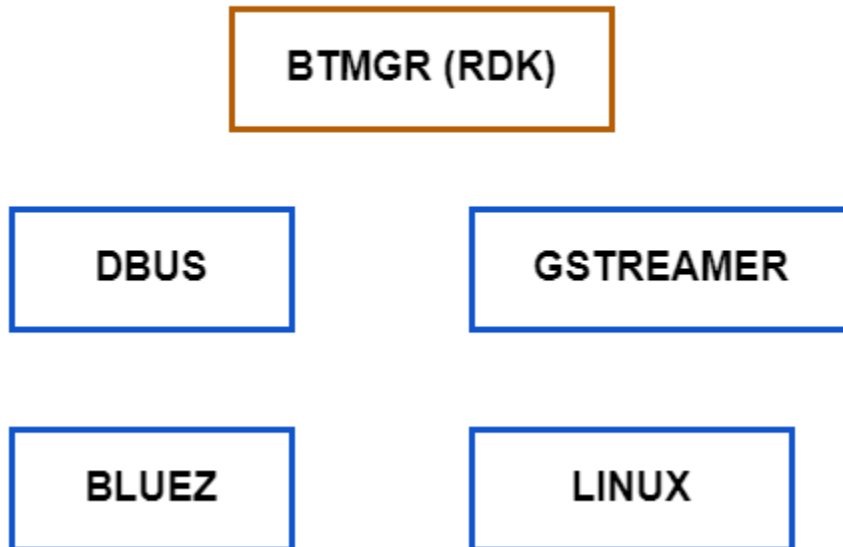
DBUS

The Bluez stack uses DBus to facilitate communication between the user application and the kernel level Bluetooth functions.

gStreamer

BTMgr utilizes gStreamer to process incoming Bluetooth data

Basic Bluetooth Architecture in RDK



Connecting to a device

In order to connect, the device must first be discovered. The source device (e.g. tablet or phone) can be put into a discoverable mode. This can be accomplished with the btrCoreTest utility. The current options are shown below:

1. Get Current Adapter
2. Scan
3. Show found devices
4. Pair
5. UnPair/Forget a device
6. Show known devices

- 7. Connect to Headset/Speakers
- 8. Disconnect to Headset/Speakers
- 9. Connect as Headset/Speakerst
- 10. Disconnect as Headset/Speakerst
- 11. Show all Bluetooth Adapters
- 12. Enable Bluetooth Adapter
- 13. Disable Bluetooth Adapter
- 14. Set Discoverable Timeout
- 15. Set Discoverable
- 16. Set friendly name
- 17. Check for audio sink capability
- 18. Check for existance of a service
- 19. Find service details
- 20. Check if Device Paired
- 21. Get Connected Dev Data path
- 22. Release Connected Dev Data path
- 23. Send SBC data to BT Headset/Speakers
- 29. BT audio input test
- 30. install agent for accepting connections NoInputNoOutput
- 31. install agent for accepting connections DisplayYesNo
- 32. Accept a connection request
- 33. Deny a connection request
- 34. Register connection callback to allow accepting or rejection of connections.
- 35. Uninstall agent - allows device-initiated pairing
- 88. debug test
- 99. Exit

Sequence of Steps

- User can use option 2 to start a device scan. The scan takes about 10 seconds to complete, then use option 3 to see the list of devices that the settop found. If your device is in the list, you can use option 4 to perform a pairing. In some cases, you may have to confirm the pairing on the tablet/phone. Once the device is paired, the next step is to establish an audio connection between the settop and your source device (e.g. tablet).
- In most cases, the connection must be initiated from the source device. Ubuntu is an exception to this, so if playing music from Ubuntu, you can start the connection from the settop. To do this, use option 9. You may have to confirm the connection on Ubuntu.
- If you are initiating the connection from the tablet, you must first start an agent. The agent allows accepting of connections from remote devices. You can do this using option 31. After invoking option 31, connect to the settop from your tablet.

Note, the agent seems to inhibit pairing from the settop, so if you need to pair more devices from the settop, you can use option 35 to unregister the agent.

- Once you have connected the device once (during the settop lifetime or with the device in persistent memory) you will not need to use the agent again. The connection will automatically be established upon attempt by Bluez, with no agent required.
- Optionally, before attempting to connect, you can run option 34 to install an authentication callback. This allows the settop to accept or refuse the connection from the tablet. If you run option 34, then attempt a first time connection from a tablet, you will be prompted to enter option 32 to accept, or 33 to deny the request.
- Once connected, you can start your player on your tablet. There is an option 29 in the btrCoreTest menu that retrieves data from Bluetooth, and pipes it into a gst-launch command to start the gstreamer elements to perform playback. However, it is not a full gstreamer implementation. To run the full gstreamer implementation, use the btrMgrStreamOutCoreTest, after first having paired and connected the device.

btrMgrStreamOutCoreTest

The current btrMgrStreamOutCoreTest having following option in RDK

- 1. Get Current Adapter
- 2. Scan
- 3. Show found devices
- 4. Pair
- 5. UnPair/Forget a device
- 6. Show known devices
- 7. Connect to Headset/Speakers
- 8. Disconnect to Headset/Speakers
- 9. Connect as Headset/Speakerst
- 10. Disconnect as Headset/Speakerst
- 11. Show all Bluetooth Adapters
- 12. Enable Bluetooth Adapter
- 13. Disable Bluetooth Adapter
- 14. Set Discoverable Timeout
- 15. Set Discoverable
- 16. Set friendly name
- 17. Check for audio sink capability
- 18. Check for existance of a service
- 19. Find service details
- 20. Check if Device Paired
- 21. Get Connected Dev Data path
- 22. Release Connected Dev Data path
- 23. Send SBC data to BT Headset/Speakers

- 24. Send WAV to BT Headset/Speakers - btrMgrStreamOutTest
- 25. Send Live Audio to BT Headset/Speakers
- 26. Send audio data from device to settop with BT
- 88. debug test
- 99. Exit

Once bluetooth connected with your tablet, start the player on your tablet, then do option 21 (this gets needed parameters for playback. Note, this functionality is automatically invoked in the btrCore utility, but has not, as of this time, been implemented in the btrMgr utility).

Once you do option 21, you can do option 26 to start Bluetooth audio, in a full gstreamer implementation.

gStreamer

Gstreamer contains several plugins for the Bluetooth In process. The gstreamer used is version 1.0.

Pipeline

```
rtpsbcdcdepay | sbcpars e | sbcdecode | brcmpcmsink
rtpsbcdcdepay: takes RTP encapsulated data from Bluetooth, and removes the RTP headers. The payload is sbc
encoded data.
```

Note it has been observed that some devices such as Ubuntu and the iPhone 5s use a constant RTP packet size, whereas some Samsung tablets have been observed to use a variable RTP packet size.

sbcpars e & sbcdecode: together, these two plugins convert the sbc data to pcm data.

brcmpcmsink: outputs pcm data to HDMI (or other source if configured, such as SPDIF out). This plugin is made by Broadcom. To date, only HDMI out has been tested.

One important note for using these utilities: Before you can use them, you must first kill a process that will be used for providing the Comcast XRE app with Bluetooth capability. You can do this as follows:

```
systemctl stop btmg r
```

Be sure to do **systemctl stop btmg r** before using the btrCore or btrMgr utilities.

API Documentation

To know more about SoC/Application level APIs details use in RDK, refer the link [BLUETOOTH MANAGER API Documentation](#)