Bluetooth Audio Router for VNC Automotive Server SDK for Android

Reference Manual

Version 4.2.4.248

http://www.vncautomotive.com/

Legal Information

See: Description

Packages

Package	Description
com.realvnc.btaudiorouter	

Legal Information

Hierarchy For All Packages

Package Hierarchies:
com.realvnc.btaudiorouter

Class Hierarchy

- java.lang.Object
  - com.realvnc.btaudiorouter.**VncBtAudioRouter**

Hierarchy For Package com.realvnc.btaudiorouter

Class Hierarchy

- java.lang.Object
  - com.realvnc.btaudiorouter.**VncBtAudioRouter**

Package com.realvnc.btaudiorouter

Class Summary

Class	Description
VncBtAudioRouter	This class provides the VNC Automotive Bluetooth Audio Routing API.

Deprecated API

Contents

Constant Field Values

Contents

com.realvnc.btaudiorouter

Class VncBtAudioRouter

java.lang.Object
    com.realvnc.btaudiorouter.VncBtAudioRouter

```
public class VncBtAudioRouter
extends java.lang.Object
```

This class provides the VNC Automotive Bluetooth Audio Routing API.

This class provides the API for allowing an Android VNC Automotive Server application to provide Bluetooth Audio Routing functionality to the VNC Automotive Viewers.

If a VNC Automotive Server application wishes to allow Bluetooth Audio Routing controlled from the Viewer side, then it has to create an instance of this class and call `register(com.realvnc.vncserver.android.VncServer)`. When the VNC Automotive session is closed, the Server application needs to call `disconnected()`.

This extension may control the Bluetooth on/off state, and the Bluetooth connections without any user intervention. This happens if the Viewer application uses the Bluetooth Audio Router for creating, or terminating a connection. If the application only checks the current connection, then no changes will be made to the Bluetooth state.

The Router is able to check the Bluetooth state, identify if a device is bonded to the mobile and find out on which audio profiles the mobile is connected to the device. The bonded state can not be checked when Bluetooth is switched off.

The Router can switch Bluetooth on silently and bond and connect to a Bluetooth audio device without user intervention. It can provide the Bluetooth PIN if needed and given by the Viewer. The Router can create connections on specific profiles and if needed it will disconnect existing connections.

The Router is also able to disconnect the mobile from a Bluetooth audio device.

Constructor Summary

Constructors

Constructor and Description
`VncBtAudioRouter(android.content.Context context)` Constructs the VncBtAudioRouter class.

Method Summary

Methods

Modifier and Type	Method and Description
`void`	`disconnected()` Notifies the Bluetooth Audio Router that the VNC Automotive session is disconnected.
`void`	`disconnectFromLastBtDevice()` Disconnects from the most recently connected Bluetooth device.
`void`	`register(com.realvnc.vncserver.android.VncServer vncServer)` Registers the VncBtAudioRouter with the VNC Automotive Server.
`java.lang.String`	`version()` Provides the version of the Router.

Methods inherited from class java.lang.Object

clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

Constructor Detail

VncBtAudioRouter

```
public VncBtAudioRouter(android.content.Context context)
```

Constructs the VncBtAudioRouter class.

Parameters:

`context` - The application context.

Method Detail

register

```
public void register(com.realvnc.vncserver.android.VncServer vncServer)
              throws com.realvnc.vncserver.core.VncException
```

Registers the VncBtAudioRouter with the VNC Automotive Server. This is the call that registers the Router with the VNC Automotive Server SDK. Once registered, if the Viewer wants to use the Bluetooth functionality on the phone it may do so. This call needs to be made before any VNC Automotive connections are started.

Parameters:

`vncServer` - The instance of the VNC Automotive Server.

Throws:

`com.realvnc.vncserver.core.VncException` - If it is not possible to register with the VNC Automotive Server.

disconnected

```
public void disconnected()
```

Notifies the Bluetooth Audio Router that the VNC Automotive session is disconnected. The VNC Automotive Server application should call this when a VNC Automotive session is disconnected.

version

```
public java.lang.String version()
```

Provides the version of the Router. This method provides the version of the Bluetooth Audio Router library.

disconnectFromLastBtDevice

```
public void disconnectFromLastBtDevice()
```

Disconnects from the most recently connected Bluetooth device.

How This API Document Is Organized

This API (Application Programming Interface) document has pages corresponding to the items in the navigation bar, described as follows.

Overview

The Overview page is the front page of this API document and provides a list of all packages with a summary for each. This page can also contain an overall description of the set of packages.

Package

Each package has a page that contains a list of its classes and interfaces, with a summary for each. This page can contain six categories:

- Interfaces (italic)
- Classes
- Enums
- Exceptions
- Errors
- Annotation Types

Class/Interface

Each class, interface, nested class and nested interface has its own separate page. Each of these pages has three sections consisting of a class/interface description, summary tables, and detailed member descriptions:

- Class inheritance diagram
- Direct Subclasses
- All Known Subinterfaces
- All Known Implementing Classes
- Class/interface declaration
- Class/interface description
- Nested Class Summary
- Field Summary
- Constructor Summary
- Method Summary
- Field Detail
- Constructor Detail
- Method Detail

Each summary entry contains the first sentence from the detailed description for that item. The summary entries are alphabetical, while the detailed descriptions are in the order they appear in the source code. This preserves the logical groupings established by the programmer.

Annotation Type

Each annotation type has its own separate page with the following sections:

- Annotation Type declaration
- Annotation Type description
- Required Element Summary
- Optional Element Summary
- Element Detail

Enum

Each enum has its own separate page with the following sections:

- Enum declaration
- Enum description
- Enum Constant Summary
- Enum Constant Detail

Tree (Class Hierarchy)

There is a Class Hierarchy page for all packages, plus a hierarchy for each package. Each hierarchy page contains a list of classes and a list of interfaces. The classes are organized by inheritance structure starting with `java.lang.Object`. The interfaces do not inherit from `java.lang.Object`.

- When viewing the Overview page, clicking on "Tree" displays the hierarchy for all packages.
- When viewing a particular package, class or interface page, clicking "Tree" displays the hierarchy for only that package.

Deprecated API

The Deprecated API page lists all of the API that have been deprecated. A deprecated API is not recommended for use, generally due to improvements, and a replacement API is usually given. Deprecated APIs may be removed in future implementations.

Index

The Index contains an alphabetic list of all classes, interfaces, constructors, methods, and fields.

Prev/Next

These links take you to the next or previous class, interface, package, or related page.

Frames/No Frames

These links show and hide the HTML frames. All pages are available with or without frames.

All Classes

The All Classes link shows all classes and interfaces except non-static nested types.

Serialized Form

Each serializable or externalizable class has a description of its serialization fields and methods. This information is of interest to re-implementors, not to developers using the API. While there is no link in the navigation bar, you can get to this information by going to any serialized class and clicking "Serialized Form" in the "See also" section of the class description.

Constant Field Values

The Constant Field Values page lists the static final fields and their values.

This help file applies to API documentation generated using the standard doclet.

C D R V

C

com.realvnc.btaudiorouter - package com.realvnc.btaudiorouter

D

disconnected() - Method in class com.realvnc.btaudiorouter.VncBtAudioRouter

Notifies the Bluetooth Audio Router that the VNC Automotive session is disconnected.

disconnectFromLastBtDevice() - Method in class com.realvnc.btaudiorouter.VncBtAudioRouter

Disconnects from the most recently connected Bluetooth device.

R

register(VncServer) - Method in class com.realvnc.btaudiorouter.VncBtAudioRouter

Registers the VncBtAudioRouter with the VNC Automotive Server.

V

version() - Method in class com.realvnc.btaudiorouter.VncBtAudioRouter

Provides the version of the Router.

VncBtAudioRouter - Class in com.realvnc.btaudiorouter

This class provides the VNC Automotive Bluetooth Audio Routing API.

VncBtAudioRouter(Context) - Constructor for class com.realvnc.btaudiorouter.VncBtAudioRouter

Constructs the VncBtAudioRouter class.

C D R V