Android.bluetooth

android.bluetooth

提供了皆如扫描设备、连接设备以及对设备间的数据传输进行管理的类，这些类对蓝牙设备进行功能性管理。

蓝牙模块 API 提供的应用包括了：

●扫描其它蓝牙设备

●通过查询本地蓝牙适配器来匹配蓝牙设备

●建立 RFCOMM（无线射频通信协议）的通道/端口

●从其他的蓝牙设备中连接到指定的端口

●传输数据到其他设备，或者从其他设备中接收数据

如需运用这些 API 来执行蓝牙通信，应用程序必须声明 BLUETOOTH 许可。对于皆如寻找设备请求等的一些附加功能，也同样需要 BLUETOOTH_ADMIN 许可。

如需要使用蓝牙 API 的更详细的指引，参看 Bluetooth Dev Guide topic.

Classes

BluetoothAdapter

代表本地的蓝牙适配器设备

BluetoothClass

代表一个描述了设备通用特性和功能的蓝牙类

BluetoothClass.Device

定义了所有设备类的常量

BluetoothClass.Device.Major

定义了所有主要设备类的常量

BluetoothClass.Service

定义了所有服务类的常量

BluetoothDevice

代表一个远程的蓝牙设备

BluetoothServerSocket

监听蓝牙服务的端口

BluetoothSocket

一个双向连接的蓝牙端口

public final class

BluetoothAdapter

extends Object

java.lang.Object

↳ android.bluetooth.BluetoothAdapter

Class Overview

代表本地的蓝牙适配器设备。BluetoothAdapter 类让用户能执行基本的蓝牙任务。例如：初始化设备的搜索，查询可匹配的设备集，使用一个已知的 MAC 地址来初始化一个 BluetoothDevice 类，创建一个 BluetoothServerSocket 类以监听其它设备对本机的连接请求等。

为了得到这个代表本地蓝牙适配器的 BluetoothAdapter 类，调用 getDefaultAdapter()这一静态方法。这是所有蓝牙动作使用的第一步。当拥有本地适配器以后，用户可以获得一系列的 BluetoothDevice 对象，这些对象代表所有拥有 getBondedDevice()方法的已经匹配的设备；用 startDiscovery()方法来开始设备的搜寻；或者创建一个 BluetoothServerSocket 类，通过 listenUsingRfcommWithServiceRecord(String, UUID)方法来监听新来的连接请求。

Note：大部分方法需要 BLUETOOTH 类的许可，一些方法同时需要 BLUETOOT_ADMIN 类的许可。

Summary

常量

String

ACTION_DISCOVERY_FINISHED

广播事件：本地蓝牙适配器已经完成设备的搜寻过程。

String

ACTION_DISCOVERY_STARTED

广播事件：本地蓝牙适配器已经开始对远程设备的搜寻过程。

String

ACTION_LOCAL_NAME_CHANGED

广播事件：本地蓝牙适配器已经更改了它的蓝牙名称。

String

ACTION_REQUEST_DISCOVERABLE

活动事件：显示一个请求被搜寻模式的系统活动。

String

ACTION_REQUEST_ENABLE

活动事件：显示一个允许用户打开蓝牙模块的系统活动。

String

ACTION_SCAN_MODE_CHANGED

广播事件：指明蓝牙扫描模块或者本地适配器已经发生变化

String

ACTION_STATE_CHANGED

广播事件：本来的蓝牙适配器的状态已经改变。

int ERROR

标记该类的错误值

String

EXTRA_DISCOVERABLE_DURATION

试图在 ACTION_REQUEST_DISCOVERABLE 常量中作为一个可选的整型附加域，来为短时间内的设备发现请求一个特定的持续时间。

String

EXTRA_LOCAL_NAME

试图在 ACTION_LOCAL_NAME_CHANGED 常量中作为一个字符串附加域，来请求本地蓝牙的名称。

String

EXTRA_PREVIOUS_SCAN_MODE

试图在 ACTION_SCAN_MODE_CHANGED 常量中作为一个整型附加域，来请求以前的扫描模式

String

EXTRA_PREVIOUS_STATE

试图在 ACTION_STATE_CHANGED 常量中作为一个整型附加域，来请求以前的供电状态。

String

EXTRA_SCAN_MODE

试图在 ACTION_SCAN_MODE_CHANGED 常量中作为一个整型附加域，来请求当前的扫描模式

String

EXTRA_STATE

试图在 ACTION_STATE_CHANGED 常量中作为一个整型附加域，来请求当前的供电状态。

int SCAN_MODE_CONNECTABLE

指明在本地蓝牙适配器中，查询扫描功能失效，但页面扫描功能有效

int SCAN_MODE_CONNECTABLE_DISCOVERABLE

指明在本地蓝牙适配器中，查询扫描功能和页面扫描功能都有效

int SCAN_MODE_NONE

指明在本地蓝牙适配器中，查询扫描功能和页面扫描功能都失效

int STATE_OFF

指明本地蓝牙适配器模块已经关闭

int STATE_ON

指明本地蓝牙适配器模块已经打开，并且准备被使用。

int STATE_TURNING_OFF

指明本地蓝牙适配器模块正在关闭

int STATE_TURNING_ON

指明本地蓝牙适配器模块正在打开

Public Methods

boolean cancelDiscovery()

.取消当前的设备发现查找进程

static boolean checkBluetoothAddress(String address)

验证皆如"00:43:A8:23:10:F0"之类的蓝牙地址，字母必须为大写才有效。

boolean disable()

关闭本地蓝牙适配器—不能在没有明确关闭蓝牙的用户动作中使用。

boolean enable()

打开本地蓝牙适配器—不能在没有明确打开蓝牙的用户动作中使用。

String

getAddress()

返回本地蓝牙适配器的硬件地址

Set<BluetoothDevice>

getBondedDevices()

返回已经匹配到本地适配器的 BluetoothDevice 类的对象集合

synchronized static BluetoothAdapter

getDefaultAdapter()

获取对默认本地蓝牙适配器的的操作权限。

String

getName()

获取本地蓝牙适配器的蓝牙呢称

BluetoothDevice

getRemoteDevice(String address)

为给予的蓝牙硬件地址获取一个 BluetoothDevice 对象。

int getScanMode()

获取本地蓝牙适配器的当前蓝牙扫描模式

int getState()

获取本地蓝牙适配器的当前状态

boolean isDiscovering()

如果当前蓝牙适配器正处于设备发现查找进程中，则返回真值

boolean isEnabled()

如果蓝牙正处于打开状态并可用，则返回真值

BluetoothServerSocket

listenUsingRfcommWithServiceRecord(String name, UUID uuid)

创建一个正在监听的安全的带有服务记录的无线射频通信蓝牙端口。

boolean setName(String name)

设置蓝牙或者本地蓝牙适配器的昵称.

boolean startDiscovery()

开始对远程设备进行查找的进程

Constants

public static final String ACTION_DISCOVERY_FINISHED

Since: API Level 5

广播事件：本地蓝牙适配器已经完成设备的搜寻过程。Requires BLUETOOTH to receive.

常量值: "android.bluetooth.adapter.action.DISCOVERY_FINISHED"

public static final String ACTION_DISCOVERY_STARTED

Since: API Level 5

广播事件：本地蓝牙适配器已经开始对远程设备的搜寻过程。

它通常牵涉到一个大概需时 12 秒的查询扫描过程，紧跟着是一个对每个获取到自身蓝牙名称的新设备的页面扫描。

Register for ACTION_FOUND to be notified as remote Bluetooth devices are found.

用户会发现一个把 ACTION_FOUND 常量通知为远程蓝牙设备的注册。

设备查找是一个重量级过程。当查找正在进行的时候，用户不能尝试对新的远程蓝牙设备进行连接，同时存在的连接将获得有限制的带宽以及高等待时间。用户可用 cencelDiscovery()类来取消正在执行的查找进程。

需要 BLUETOOTH 类来接收。

常量名："android.bluetooth.adapter.action.DISCOVERY_STARTED"

public static final String ACTION_LOCAL_NAME_CHANGED

Since: API Level 5

广播事件：本地蓝牙适配器已经更改了它的蓝牙名称。

该名称对远程蓝牙设备是可见的。

它总是包含了一个带有名称的 EXTRA_LOCAL_NAME 附加域。

它需要请求 BLUETOOTH 类去获取这个名字。

常量值: "android.bluetooth.adapter.action.LOCAL_NAME_CHANGED"

public static final String ACTION_REQUEST_DISCOVERABLE

Since: API Level 5

活动事件：显示一个请求被搜寻模式的系统活动。如果蓝牙模块当前未打开，该活动也将请求用户打开蓝牙模块。

被搜寻模式和 SCAN_MODE_CONNECTABLE_DISCOVERABLE 等价。当远程设备执行查找进程的时候，它允许其发现该蓝牙适配器。

从隐私安全考虑，Android 不会将被搜寻模式设置为默认状态。

该意图的发送者可以选择性地运用 EXTRA_DISCOVERABLE_DURATION 这个附加域去请求发现设备的持续时间。普遍来说，对于每一请求，默认的持续时间为 120 秒，最大值则可达到 300 秒。

Android 运用 onActivityResult(int, int, Intent)回收方法来传递该活动结果的通知。被搜寻的时间（以秒为单位）将通过 resultCode 值来显示，如果用户拒绝被搜寻，或者设备产生了错误，则通过 RESULT_CANCELED 值来显示。

每当扫描模式变化的时候，应用程序可以为通过 ACTION_SCAN_MODE_CHANGED 值来监听全局的消息通知。比如，当设备停止被搜寻以后，该消息可以被系统通知給应用程序。

需要 BLUETOOTH 类

常量值: "android.bluetooth.adapter.action.REQUEST_DISCOVERABLE"

public static final String ACTION_REQUEST_ENABLE

Since: API Level 5

活动事件：显示一个允许用户打开蓝牙模块的系统活动。

当蓝牙模块完成打开工作，或者当用户决定不打开蓝牙模块时，系统活动将返回该值。

Android 运用 onActivityResult(int, int, Intent)回收方法来传递该活动结果的通知。如果蓝牙模块被打开，将通过 resultCode 值来显示；如果用户拒绝该请求，或者设备产生了错误，则通过 RESULT_CANCELED 值来显示。

每当蓝牙模块被打开或者关闭，应用程序可以为通过 ACTION_SCAN_MODE_CHANGED 值来监听全局的消息通知。

需要 BLUETOOTH 类

常量值: "android.bluetooth.adapter.action.REQUEST_ENABLE"

public static final String ACTION_SCAN_MODE_CHANGED

Since: API Level 5

广播事件：指明蓝牙扫描模块或者本地适配器已经发生变化

它总是包含 EXTRA_SCAN_MODE 和 EXTRA_PREVIOUS_SCAN_MODE。这两个附加域各自包含了新的和旧的扫描模式。

需要 BLUETOOTH 类

常量值: "android.bluetooth.adapter.action.SCAN_MODE_CHANGED"

public static final String ACTION_STATE_CHANGED

Since: API Level 5

广播事件：本来的蓝牙适配器的状态已经改变。

比如：蓝牙模块已经被打开或者关闭。

它总是包含 EXTRA_STATE 和 EXTRA_PREVIOUS_STATE。这两个附加域各自包含了新的和旧的状态。

需要 BLUETOOTH 类去接收

常量值: "android.bluetooth.adapter.action.STATE_CHANGED"

public static final int ERROR

Since: API Level 5

标记该类的错误值。确保和该类中的任意其它整数常量不相等。它为需要一个标记错误值的函数提供了便利。例如： Intent.getIntExtra(BluetoothAdapter.EXTRA_STATE, BluetoothAdapter.ERROR)

常量值: -2147483648 (0x80000000)

public static final String EXTRA_DISCOVERABLE_DURATION

Since: API Level 5

试图在 ACTION_REQUEST_DISCOVERABLE 常量中作为一个可选的整型附加域，来为短时间内的设备发现请求一个特定的持续时间。默认值为 120 秒，超过 300 秒的请求将被限制。这些值是可以变化的。

常量值: "android.bluetooth.adapter.extra.DISCOVERABLE_DURATION"

public static final String EXTRA_LOCAL_NAME

Since: API Level 5

试图在 ACTION_LOCAL_NAME_CHANGED 常量中作为一个字符串附加域，来请求本地蓝牙的名称。

常量值: "android.bluetooth.adapter.extra.LOCAL_NAME"

public static final String EXTRA_PREVIOUS_SCAN_MODE

Since: API Level 5

试图在 ACTION_SCAN_MODE_CHANGED 常量中作为一个整型附加域，来请求以前的扫描模式。可能值有: SCAN_MODE_NONE, SCAN_MODE_CONNECTABLE,

SCAN_MODE_CONNECTABLE_DISCOVERABLE,

常量值: "android.bluetooth.adapter.extra.PREVIOUS_SCAN_MODE"

public static final String EXTRA_PREVIOUS_STATE

Since: API Level 5

试图在 ACTION_STATE_CHANGED 常量中作为一个整型附加域，来请求以前的供电状态。 可能值有: STATE_OFF, STATE_TURNING_ON, STATE_ON, STATE_TURNING_OFF,

常量值: "android.bluetooth.adapter.extra.PREVIOUS_STATE"

public static final String EXTRA_SCAN_MODE

Since: API Level 5

试图在 ACTION_SCAN_MODE_CHANGED 常量中作为一个整型附加域，来请求当前的扫描模式.可能值有: SCAN_MODE_NONE, SCAN_MODE_CONNECTABLE,

SCAN_MODE_CONNECTABLE_DISCOVERABLE,

常量值: "android.bluetooth.adapter.extra.SCAN_MODE"

public static final String EXTRA_STATE

Since: API Level 5

试图在 ACTION_STATE_CHANGED 常量中作为一个整型附加域，来请求当前的供电状态。 可能值有: STATE_OFF, STATE_TURNING_ON, STATE_ON, STATE_TURNING_OFF,

常量值: "android.bluetooth.adapter.extra.STATE"

public static final int SCAN_MODE_CONNECTABLE

Since: API Level 5

指明在本地蓝牙适配器中，查询扫描功能失效，但页面扫描功能有效。因此该设备不能被远程蓝牙设备发现，但如果以前曾经发现过该设备，则远程设备可以对其进行连接。

常量值: 21 (0x00000015)

public static final int SCAN_MODE_CONNECTABLE_DISCOVERABLE

Since: API Level 5

指明在本地蓝牙适配器中，查询扫描功能和页面扫描功能都有效。因此该设备既可以被远程蓝牙设备发现，也可以被其连接。

常量值: 23 (0x00000017)

public static final int SCAN_MODE_NONE

Since: API Level 5

指明在本地蓝牙适配器中，查询扫描功能和页面扫描功能都失效. 因此该设备既不可以被远程蓝牙设备发现，也不可以被其连接。

常量值: 20 (0x00000014)

public static final int STATE_OFF

Since: API Level 5

指明本地蓝牙适配器模块已经关闭

常量值: 10 (0x0000000a)

public static final int STATE_ON

Since: API Level 5

指明本地蓝牙适配器模块已经打开，并且准备被使用。

常量值: 12 (0x0000000c)

public static final int STATE_TURNING_OFF

Since: API Level 5

指明本地蓝牙适配器模块正在关闭。本地客户端可以立刻尝试友好地断开任意外部连接。

常量值: 13 (0x0000000d)

public static final int STATE_TURNING_ON

Since: API Level 5

指明本地蓝牙适配器模块正在打开. 然而本地客户在尝试使用这个适配器之前需要为 STATE_ON 状态而等待。

常量值: 11 (0x0000000b)

Public Methods

public boolean cancelDiscovery ()

Since: API Level 5

.取消当前的设备发现查找进程

需要 BLUETOOTH_ADMIN 类

因为对蓝牙适配器而言，查找是一个重量级的过程，因此这个方法必须在尝试连接到远程设备前使用用connect()方法进行调用。发现的过程不会由活动来进行管理，但是它会作为一个系统服务来运行，因此即使它不能直接请求这样的一个查询动作，也必需取消该搜索进程。

返回值：

成功则返回 true，有错误则返回 false。

public static boolean checkBluetoothAddress (String address)

Since: API Level 5

验证皆如"00:43:A8:23:10:F0"之类的蓝牙地址，字母必须为大写才有效。

参数

地址 字符串形式的蓝牙模块地址

返回

地址正确则返回 true，否则返回 false。

public boolean disable ()

Since: API Level 5

关闭本地蓝牙适配器—不能在没有明确关闭蓝牙的用户动作中使用。

这个方法友好地停止所有的蓝牙连接，停止蓝牙系统服务，以及对所有基础蓝牙硬件进行断电。

没有用户的直接同意，蓝牙永远不能被禁止。这个 disable()方法只提供了一个应用，该应用包含了一个改变系统设置的用户界面（例如“电源控制”应用）。

这是一个异步调用方法：该方法将马上获得返回值，用户要通过监听 ACTION_STATE_CHANGED 值来获取随后的适配器状态改变的通知。如果该调用返回 true 值，则该适配器状态会立刻从 STATE_ON 转向 STATE_TURNING_OFF，稍后则会转为 STATE_OFF 或者 STATE_ON。如果该调用返回 false，那么系统已经有一个保护蓝牙适配器被关闭的问题—比如该适配器已经被关闭了。

需要 BLUETOOTH_ADMIN 类的许可

返回值

如果蓝牙适配器的停止进程已经开启则返回 true，如果产生错误则返回 false。

public boolean enable ()

Since: API Level 5

打开本地蓝牙适配器—不能在没有明确打开蓝牙的用户动作中使用。

该方法将为基础的蓝牙硬件供电，并且启动所有的蓝牙系统服务。

没有用户的直接同意，蓝牙永远不能被禁止。如果用户为了创建无线连接而打开了蓝牙模块，则其需要 ACTION_REQUEST_ENABLE 值，该值将提出一个请求用户允许以打开蓝牙模块的会话。这个 enable() 值只提供了一个应用，该应用包含了一个改变系统设置的用户界面（例如“电源控制”应用）。

这是一个异步调用方法：该方法将马上获得返回值，用户要通过监听 ACTION_STATE_CHANGED 值来获取随后的适配器状态改变的通知。如果该调用返回 true 值，则该适配器状态会立刻从 STATE_OFF 转向 STATE_TURNING_ON，稍后则会转为 STATE_OFF 或者 STATE_ON。如果该调用返回 false，那么说明系统已经有一个保护蓝牙适配器被打开的问题—比如飞行模式，或者该适配器已经被打开。

需要 BLUETOOTH_ADMIN 类的许可

如果蓝牙适配器的打开进程已经开启则返回 true，如果产生错误则返回 false。

public String getAddress ()

Since: API Level 5

返回本地蓝牙适配器的硬件地址

例如： "00:11:22:AA:BB:CC".

需要 BLUETOOTH 类

返回值

字符串形式的蓝牙模块地址

public Set<BluetoothDevice> getBondedDevices ()

Since: API Level 5

返回已经匹配到本地适配器的 BluetoothDevice 类的对象集合

需要 BLUETOOTH 类

返回值

未被修改的 BluetoothDevice 类的对象集合，如果有错误则返回 null。

public static synchronized BluetoothAdapter getDefaultAdapter ()

Since: API Level 5

获取对默认本地蓝牙适配器的的操作权限。

目前 Andoird 只支持一个蓝牙适配器，但是 API 可以被扩展为支持多个适配器。该方法总是返回默认的适配器。

返回值

返回默认的本地适配器，如果蓝牙适配器在该硬件平台上不能被支持，则返回 null。

public String getName ()

Since: API Level 5

获取本地蓝牙适配器的蓝牙呢称

这个呢称对于外界蓝牙设备而言是可见的。

需要 BLUETOOTH 类

返回值

该蓝牙适配器名称，如果有错误则返回 null

public BluetoothDevice getRemoteDevice (String address)

Since: API Level 5

为给予的蓝牙硬件地址获取一个 BluetoothDevice 对象。

Valid Bluetooth hardware addresses must be upper case, in a format such as "00:11:22:33:AA:BB". The helper checkBluetoothAddress(String) is available to validate a Bluetooth address.

合法的蓝牙硬件地址必须为大写，格式类似于"00:11:22:33:AA:BB"。checkBluetoothAddress(String)方法可以用来验证蓝牙地址的正确性。

A BluetoothDevice will always be returned for a valid hardware address, even if this adapter has never seen that device.

BluetoothDevice 类对于合法的硬件地址总会产生返回值，即使这个适配器从未见过该设备。

参数

地址 合法的蓝牙 MAC 地址

异常抛出

IllegalArgumentException

如果地址不合法

public int getScanMode ()

Since: API Level 5

获取本地蓝牙适配器的当前蓝牙扫描模式

蓝牙扫描模式决定本地适配器可连接并且/或者可被远程蓝牙设备所连接。

可能值有: SCAN_MODE_NONE, SCAN_MODE_CONNECTABLE, SCAN_MODE_CONNECTABLE_DISCOVERABLE.

需要 BLUETOOTH 类

返回值

扫描模式

public int getState ()

Since: API Level 5

获取本地蓝牙适配器的当前状态

可能值有 STATE_OFF, STATE_TURNING_ON, STATE_ON, STATE_TURNING_OFF.

需要 BLUETOOTH 类

返回值

蓝牙适配器的当前状态

public boolean isDiscovering ()

Since: API Level 5

如果当前蓝牙适配器正处于设备发现查找进程中，则返回真值

设备查找是一个重量级过程。当查找正在进行的时候，用户不能尝试对新的远程蓝牙设备进行连接，同时存在的连接将获得有限制的带宽以及高等待时间。用户可用 cencelDiscovery()类来取消正在执行的查找进程。

应用程序也可以为 ACTION_DISCOVERY_STARTED 或者 ACTION_DISCOVERY_FINISHED 进行注册，从而当查找开始或者完成的时候，可以获得通知。

需要 BLUETOOTH 类

返回值

如果正在查找，则返回 true

public boolean isEnabled ()

Since: API Level 5

如果蓝牙正处于打开状态并可用，则返回真值

和 getBluetoothState()==STATE_ON 等价

需要 BLUETOOTH 类

返回值

如果本地适配器已经打开，则返回 true

public BluetoothServerSocket listenUsingRfcommWithServiceRecord (String name, UUID uuid)

Since: API Level 5

创建一个正在监听的安全的带有服务记录的无线射频通信（RFCOMM）蓝牙端口。

一个对该端口进行连接的远程设备将被认证，对该端口的通讯将被加密。

使用 accpet（）方法可以获取从监听 BluetoothServerSocket 处新来的连接

该系统分配一个未被使用的无线射频通信通道来进行监听。

The system will also register a Service Discovery Protocol (SDP) record with the local SDP server containing the specified UUID, service name, and auto-assigned channel. Remote Bluetooth devices can use the same UUID to query our SDP server and discover which channel to connect to. This SDP record

will be removed when this socket is closed, or if this application closes unexpectedly.

该系统也将注册一个服务探索协议（SDP）记录，该记录带有一个包含了特定的通用唯一识别码（Universally Unique Identifier，UUID），服务器名称和自动分配通道的本地 SDP 服务。远程蓝牙设备可以用相同的 UUID 来查询自己的 SDP 服务器，并搜寻连接到了哪个通道上。如果该端口已经关闭，或者如果该应用程序异常退出，则这个 SDP 记录会被移除。

Use createRfcommSocketToServiceRecord(UUID) to connect to this socket from another device using the same UUID.

使用 createRfcommSocketToServiceRecord(UUID)从另一使用相同 UUID 的设备来连接到这个端口

需要 BLUETOOTH 类

Parameters

名字 SDP 记录下的服务器名

UUID SDP 记录下的 UUID

返回值

一个正在监听的无线射频通信蓝牙服务端口

抛出异常

IOException

产生错误，比如蓝牙设备不可用，或者许可无效，或者通道被占用。

public boolean setName (String name)

Since: API Level 5

设置蓝牙或者本地蓝牙适配器的昵称.

这个名字对于外界蓝牙设备而言是可见的。

Valid Bluetooth names are a maximum of 248 UTF-8 characters, however many remote devices can only display the first 40 characters, and some may be limited to just 20.

合法的蓝牙名称最多拥有 248 位 UTF-8 字符，但是很多外界设备只能显示前 40 个字符，有些可能只限制前 20 个字符。

需要 BLUETOOTH_ADMIN 类

参数

名称 一个合法的蓝牙名称

返回值

如果该名称已被设定，则返回 true，否则返回 false

public boolean startDiscovery ()

Since: API Level 5

开始对远程设备进行查找的进程

它通常牵涉到一个大概需时 12 秒的查询扫描过程，紧跟着是一个对每个获取到自身蓝牙名称的新设备的页面扫描。

这是一个异步调用方法：该方法将马上获得返回值，注册 ACTION_DISCOVERY_STARTED and ACTION_DISCOVERY_FINISHED 意图准确地确定该探索是处于开始阶段或者完成阶段。注册 ACTION_FOUND 以活动远程蓝牙设备已找到的通知。

设备查找是一个重量级过程。当查找正在进行的时候，用户不能尝试对新的远程蓝牙设备进行连接，同时存在的连接将获得有限制的带宽以及高等待时间。用户可用 cencelDiscovery()类来取消正在执行的查找进程。发现的过程不会由活动来进行管理，但是它会作为一个系统服务来运行，因此即使它不能直接请求这样的一个查询动作，也必需取消该搜索进程。

Device discovery will only find remote devices that are currently discoverable (inquiry scan enabled).

Many Bluetooth devices are not discoverable by default, and need to be entered into a special mode.

设备搜寻只寻找已经被连接的远程设备。许多蓝牙设备默认不会被搜寻到，并且需要进入到一个特殊的模式当中。

需要 BLUETOOTH_ADMIN 类

返回值

成功则返回 true，有错误则返回 false。

public final class

BluetoothClass

extends Object

implements Parcelable

java.lang.Object

↳ android.bluetooth.BluetoothClass

Class Overview

代表一个描述了设备通用特性和功能的蓝牙类。比如，一个蓝牙类会指定皆如电话、计算机或耳机的通用设备类型，可以提供皆如音频或者电话的服务。

每个蓝牙类都是有 0 个或更多的服务类，以及一个设备类组成。设备类将被分解成主要和较小的设备类部分。

BluetoothClass 用作一个能粗略描述一个设备（比如关闭用户界面上一个图标的设备）的线索，但当蓝牙服务事实上是被一个设备所支撑的时候，BluetoothClass 的介绍则不那么可信任。精确的服务搜寻通过 SDP 请求来完成。当运用 createRfcommSocketToServiceRecord(UUID) 和 listenUsingRfcommWithServiceRecord(String, UUID)来创建 RFCOMM 端口的时候，SDP 请求就会自动

执行。

使用 getBluetoothClass()方法来获取为远程设备所提供的类。

Summary

嵌套类

class BluetoothClass.Device

定义所有设备类的常量

class BluetoothClass.Service

定义所有服务类的常量

常量

Creator<BluetoothClass>

CREATOR

继承常量

From interface android.os.Parcelable

int CONTENTS_FILE_DESCRIPTOR

使用 describeContents()方法的位屏蔽：每个位代表一种在数据流序列中被认为具有特殊意义的对象

int PARCELABLE_WRITE_RETURN_VALUE

使用 writeToParcel(Parcel, int)方法的标志：这个被写入的对象是一个返回值，它是一个皆如"Parcelable someFunction()", "void someFunction(out Parcelable)", or "void someFunction(inout Parcelable)"等函数的结果。

公共方法

int describeContents()

描述了包含在 Parcelable's marshalled representation 中的特殊对象的种类。

boolean equals(Object o)

比较带有特定目标的常量。如果他们相等则标示出来。

int getDeviceClass()

返回 BluetoothClass.中的设备类部分（主要的和较小的）

int getMajorDeviceClass()

返回 BluetoothClass.中设备类的主要部分

boolean hasService(int service)

如果该指定服务类被 BluetoothClass.所支持，则返回 true

int hashCode()

返回这个对象的整型哈希码

String

toString()

返回这个对象的字符串，该字符串包含精确且可读的介绍

void writeToParcel(Parcel out, int flags)

将类的数据写入外部提供的 Parcel 中（注：此处没有按照原文翻译，实在看不懂原文的意思，囧）

继承方法

From class java.lang.Object

Object

clone()

创建并返回该对象的复制品

boolean equals(Object o)

比较带有特定目标的常量。如果他们相等则标示出来。

void finalize()

对象的内存被虚拟机收回前需要调用该方法

final Class<? extends Object>

getClass()

返回一个唯一的 Class 常量，该常量代表这个对象类

int hashCode()

返回这个对象的整型哈希码

final void notify()

产生一个在该对象监测器中等待被唤醒（通过 wait() 方法进行调用）的线程

final void notifyAll()

产生所有在该对象监测器中等待被唤醒（通过 wait() 方法进行调用）的线程

String

toString()

返回这个对象的字符串，该字符串包含精确且可读的介绍

final void wait(long millis, int nanos)

产生一个调用线程，该线程将等待，直到另一个线程调用了 notify()或 notifyAll()方法，或者直到产生一个精确的超时。

final void wait(long millis)

产生一个调用线程，该线程将等待，直到另一个线程调用了 notify()或 notifyAll()方法，或者直到产生一个精确的超时。

final void wait()

产生一个调用线程，该线程将等待，直到另一个线程调用了该对象中的 notify()或 notifyAll()方法

From interface android.os.Parcelable

abstract int describeContents()

描述了包含在 Parcelable's marshalled representation.中的特殊对象的种类。

abstract void writeToParcel(Parcel dest, int flags)

将类的数据写入外部提供的 Parcel 中

Constants

public static final Creator<BluetoothClass> CREATOR

Since: API Level 5

Public Methods

public int describeContents ()

Since: API Level 5

描述了包含在 Parcelable's marshalled representation.中的特殊对象的种类。

返回值

一个指示被 Parcelabel 所排列的特殊对象类型集合的位屏蔽。

public boolean equals (Object o)

Since: API Level 5

比较带有特定目标的常量。如果他们相等则标示出来。 为了保证其相等，o 必须代表相同的对象，该对象作为这个使用类依赖比较的常量。通常约定，该比较既要可移植又需灵活

当且仅当 o 是一个作为接收器（使用==操作符来做比较）的精确相同的对象是，这个对象的实现才返回 true 值。子类通常实现 equals(Object)方法，这样它才会重视这两个对象的类型和状态。

通常约定，对于 equals(Object)和 hashCode() 方法，如果 equals 对于任意两个对象返回真值，那么 hashCode()必须对这些对象返回相同的纸。这意味着对象的子类通常都覆盖或者都不覆盖这两个方法。

参数

o 需要对比常量的对象

返回值

如果特定的对象和该对象相等则返回 true，否则返回 false。

public int getDeviceClass ()

Since: API Level 5

返回 BluetoothClass.中的设备类部分（主要的和较小的）

从函数中返回的值可以和在 BluetoothClass.Device 中的公共常量做比较，从而确定哪个设备类在这个蓝牙类中是被编码的。

返回值

设备类部分

public int getMajorDeviceClass ()

Since: API Level 5

返回 BluetoothClass.中设备类的主要部分

从函数中返回的值可以和在 BluetoothClass.Device.Major 中的公共常量做比较，从而确定哪个主要类在这个蓝牙类中是被编码的。

返回值

主要设备类部分

public boolean hasService (int service)

Since: API Level 5

如果该指定服务类被 BluetoothClass.所支持，则返回 true

在 BluetoothClass.Service 中，合法的服务类是公共常量，比如 AUDIO.类。

参数

服务 合法服务类

返回值

如果该服务类可被支持，则返回 true

public int hashCode ()

Since: API Level 5

返回这个对象的整型哈希码。按约定，任意两个在 equals(Object)中返回 true 的对象必须返回相同的哈希码。这意味着对象的子类通常通常覆盖或者都不覆盖这两个方法。

返回值

该对象的哈希码

public String toString ()

Since: API Level 5

返回这个对象的字符串，该字符串包含精确且可读的介绍。系统鼓励子类去重写该方法，并且提供了能对该对象的类型和数据进行重视的实现方法。默认的实现方法只是简单地把类名、“@“符号和该对象 hashCode()方法的 16 进制数连接起来（如下列所示的表达式）：

getClass().getName() + '@' + Integer.toHexString(hashCode())

返回值

该对象中一个可被打印的字符串。

public void writeToParcel (Parcel out, int flags)

Since: API Level 5

将类的数据写入外部提供的 Parcel 中

Parameters

out 对象需要被写入的 Parcel

flags 和对象需要如何被写入有关的附加标志。可能是 0，或者可能是 PARCELABLE_WRITE_RETURN_VALUE。

public static class

BluetoothClass.Device

extends Object

java.lang.Object

↳ android.bluetooth.BluetoothClass.Device

Class Overview

定义所有的设备类常量

每个 BluetoothClass 对一个带有主要和较小部分的设备类进行编码。

BluetoothClass.Device 中的常量代表主要和较小的设备类部分（完整的设备类）的组合。

BluetoothClass.Device.Major 的常量只能代表主要设备类。

Summary

嵌套类

class BluetoothClass.Device.Major

定义了所有的主要设备类常量

Constants

int AUDIO_VIDEO_CAMCORDER

int AUDIO_VIDEO_CAR_AUDIO

int AUDIO_VIDEO_HANDSFREE

int AUDIO_VIDEO_HEADPHONES

int AUDIO_VIDEO_HIFI_AUDIO

int AUDIO_VIDEO_LOUDSPEAKER

int AUDIO_VIDEO_MICROPHONE

int AUDIO_VIDEO_PORTABLE_AUDIO

int AUDIO_VIDEO_SET_TOP_BOX

int AUDIO_VIDEO_UNCATEGORIZED

int AUDIO_VIDEO_VCR

int AUDIO_VIDEO_VIDEO_CAMERA

int AUDIO_VIDEO_VIDEO_CONFERENCING

int AUDIO_VIDEO_VIDEO_DISPLAY_AND_LOUDSPEAKER

int AUDIO_VIDEO_VIDEO_GAMING_TOY

int AUDIO_VIDEO_VIDEO_MONITOR

int AUDIO_VIDEO_WEARABLE_HEADSET

int COMPUTER_DESKTOP

int COMPUTER_HANDHELD_PC_PDA

int COMPUTER_LAPTOP

int COMPUTER_PALM_SIZE_PC_PDA

int COMPUTER_SERVER

int COMPUTER_UNCATEGORIZED

int COMPUTER_WEARABLE

int HEALTH_BLOOD_PRESSURE

int HEALTH_DATA_DISPLAY

int HEALTH_GLUCOSE

int HEALTH_PULSE_OXIMETER

int HEALTH_PULSE_RATE

int HEALTH_THERMOMETER

int HEALTH_UNCATEGORIZED

int HEALTH_WEIGHING

int PHONE_CELLULAR

int PHONE_CORDLESS

int PHONE_ISDN

int PHONE_MODEM_OR_GATEWAY

int PHONE_SMART

int PHONE_UNCATEGORIZED

int TOY_CONTROLLER

int TOY_DOLL_ACTION_FIGURE

int TOY_GAME

int TOY_ROBOT

int TOY_UNCATEGORIZED

int TOY_VEHICLE

int WEARABLE_GLASSES

int WEARABLE_HELMET

int WEARABLE_JACKET

int WEARABLE_PAGER

int WEARABLE_UNCATEGORIZED

int WEARABLE_WRIST_WATCH

公共构造器

BluetoothClass.Device()

继承方法

From class java.lang.Object

Object

clone()

创建并返回该对象的复制品.

boolean equals(Object o)

比较带有特定目标的常量。如果他们相等则标示出来。

void finalize()

对象的内存被虚拟机收回前需要调用该方法

final Class<? extends Object>

getClass()

返回一个唯一的 Class 常量，该常量代表这个对象类

int hashCode()

返回该对象的一个整型哈希码

final void notify()

产生一个在该对象监测器中等待被唤醒（通过 wait() 方法进行调用）的线程

final void notifyAll()

产生在对象监测器中等待被唤醒（通过 wait() 方法进行调用）的所有线程

String

toString()

返回这个对象的字符串，该字符串包含精确且可读的介绍

final void wait(long millis, int nanos)

产生一个调用线程，该线程将等待，直到另一个线程调用了 notify()或 notifyAll()方法，或者直到产生一个精确的超时。

final void wait(long millis)

产生一个调用线程，该线程将等待，直到另一个线程调用了 notify()或 notifyAll()方法，或者直到产生一个精确的超时。

final void wait()

产生一个调用线程，该线程将等待，直到另一个线程调用了在该对象中的 notify()或 notifyAll()方法。

Constants

public static final int AUDIO_VIDEO_CAMCORDER

Since: API Level 5

常量值: 1076 (0x00000434)

public static final int AUDIO_VIDEO_CAR_AUDIO

Since: API Level 5

常量值: 1056 (0x00000420)

public static final int AUDIO_VIDEO_HANDSFREE

Since: API Level 5

常量值: 1032 (0x00000408)

public static final int AUDIO_VIDEO_HEADPHONES

Since: API Level 5

常量值: 1048 (0x00000418)

public static final int AUDIO_VIDEO_HIFI_AUDIO

Since: API Level 5

常量值: 1064 (0x00000428)

public static final int AUDIO_VIDEO_LOUDSPEAKER

Since: API Level 5

常量值: 1044 (0x00000414)

public static final int AUDIO_VIDEO_MICROPHONE

Since: API Level 5

常量值: 1040 (0x00000410)

public static final int AUDIO_VIDEO_PORTABLE_AUDIO

Since: API Level 5

常量值: 1052 (0x0000041c)

public static final int AUDIO_VIDEO_SET_TOP_BOX

Since: API Level 5

常量值: 1060 (0x00000424)

public static final int AUDIO_VIDEO_UNCATEGORIZED

Since: API Level 5

常量值: 1024 (0x00000400)

public static final int AUDIO_VIDEO_VCR

Since: API Level 5

常量值: 1068 (0x0000042c)

public static final int AUDIO_VIDEO_VIDEO_CAMERA

Since: API Level 5

常量值: 1072 (0x00000430)

public static final int AUDIO_VIDEO_VIDEO_CONFERENCING

Since: API Level 5

常量值: 1088 (0x00000440)

public static final int AUDIO_VIDEO_VIDEO_DISPLAY_AND_LOUDSPEAKER

Since: API Level 5

常量值: 1084 (0x0000043c)

public static final int AUDIO_VIDEO_VIDEO_GAMING_TOY

Since: API Level 5

常量值: 1096 (0x00000448)

public static final int AUDIO_VIDEO_VIDEO_MONITOR

Since: API Level 5

常量值: 1080 (0x00000438)

public static final int AUDIO_VIDEO_WEARABLE_HEADSET

Since: API Level 5

常量值: 1028 (0x00000404)

public static final int COMPUTER_DESKTOP

Since: API Level 5

常量值: 260 (0x00000104)

public static final int COMPUTER_HANDHELD_PC_PDA

Since: API Level 5

常量值: 272 (0x00000110)

public static final int COMPUTER_LAPTOP

Since: API Level 5

常量值: 268 (0x0000010c)

public static final int COMPUTER_PALM_SIZE_PC_PDA

Since: API Level 5

常量值: 276 (0x00000114)

public static final int COMPUTER_SERVER

Since: API Level 5

常量值: 264 (0x00000108)

public static final int COMPUTER_UNCATEGORIZED

Since: API Level 5

常量值: 256 (0x00000100)

public static final int COMPUTER_WEARABLE

Since: API Level 5

常量值: 280 (0x00000118)

public static final int HEALTH_BLOOD_PRESSURE

Since: API Level 5

常量值: 2308 (0x00000904)

public static final int HEALTH_DATA_DISPLAY

Since: API Level 5

常量值: 2332 (0x0000091c)

public static final int HEALTH_GLUCOSE

Since: API Level 5

常量值: 2320 (0x00000910)

public static final int HEALTH_PULSE_OXIMETER

Since: API Level 5

常量值: 2324 (0x00000914)

public static final int HEALTH_PULSE_RATE

Since: API Level 5

常量值: 2328 (0x00000918)

public static final int HEALTH_THERMOMETER

Since: API Level 5

常量值: 2312 (0x00000908)

public static final int HEALTH_UNCATEGORIZED

Since: API Level 5

常量值: 2304 (0x00000900)

public static final int HEALTH_WEIGHING

Since: API Level 5

常量值: 2316 (0x0000090c)

public static final int PHONE_CELLULAR

Since: API Level 5

常量值: 516 (0x00000204)

public static final int PHONE_CORDLESS

Since: API Level 5

常量值: 520 (0x00000208)

public static final int PHONE_ISDN

Since: API Level 5

常量值: 532 (0x00000214)

public static final int PHONE_MODEM_OR_GATEWAY

Since: API Level 5

常量值: 528 (0x00000210)

public static final int PHONE_SMART

Since: API Level 5

常量值: 524 (0x0000020c)

public static final int PHONE_UNCATEGORIZED

Since: API Level 5

常量值: 512 (0x00000200)

public static final int TOY_CONTROLLER

Since: API Level 5

常量值: 2064 (0x00000810)

public static final int TOY_DOLL_ACTION_FIGURE

Since: API Level 5

常量值: 2060 (0x0000080c)

public static final int TOY_GAME

Since: API Level 5

常量值: 2068 (0x00000814)

public static final int TOY_ROBOT

Since: API Level 5

常量值: 2052 (0x00000804)

public static final int TOY_UNCATEGORIZED

Since: API Level 5

常量值: 2048 (0x00000800)

public static final int TOY_VEHICLE

Since: API Level 5

常量值: 2056 (0x00000808)

public static final int WEARABLE_GLASSES

Since: API Level 5

常量值: 1812 (0x00000714)

public static final int WEARABLE_HELMET

Since: API Level 5

常量值: 1808 (0x00000710)

public static final int WEARABLE_JACKET

Since: API Level 5

常量值: 1804 (0x0000070c)

public static final int WEARABLE_PAGER

Since: API Level 5

常量值: 1800 (0x00000708)

public static final int WEARABLE_UNCATEGORIZED

Since: API Level 5

常量值: 1792 (0x00000700)

public static final int WEARABLE_WRIST_WATCH

Since: API Level 5

常量值: 1796 (0x00000704)

public static class

BluetoothClass.Device.Major

extends Object

java.lang.Object

↳ android.bluetooth.BluetoothClass.Device.Major

Class Overview

定义所有的主要设备类函数

Summary

常量

int AUDIO_VIDEO

int COMPUTER

int HEALTH

int IMAGING

int MISC

int NETWORKING

int PERIPHERAL

int PHONE

int TOY

int UNCATEGORIZED

int WEARABLE

公共构造器

BluetoothClass.Device.Major()

继承方法

From class java.lang.Object

Object

clone()

创建并返回该对象的复制品.

boolean equals(Object o)

比较带有特定目标的常量。如果他们相等则标示出来。

void finalize()

对象的内存被虚拟机收回前需要调用该方法

final Class<? extends Object>

getClass()

返回一个唯一的 Class 常量，该常量代表这个对象类

int hashCode()

返回该对象的一个整型哈希码

final void notify()

产生一个在该对象监测器中等待被唤醒（通过 wait() 方法进行调用）的线程

final void notifyAll()

产生在对象监测器中等待被唤醒（通过 wait() 方法进行调用）的所有线程

String

toString()

返回这个对象的字符串，该字符串包含精确且可读的介绍

final void wait(long millis, int nanos)

产生一个调用线程，该线程将等待，直到另一个线程调用了 notify()或 notifyAll()方法，或者直到产生一个精确的超时。

final void wait(long millis)

产生一个调用线程，该线程将等待，直到另一个线程调用了 notify()或 notifyAll()方法，或者直到产生一个精确的超时。

final void wait()

产生一个调用线程，该线程将等待，直到另一个线程调用了在该对象中的 notify()或 notifyAll()方法。

Constants

public static final int AUDIO_VIDEO

Since: API Level 5

常量值: 1024 (0x00000400)

public static final int COMPUTER

Since: API Level 5

常量值: 256 (0x00000100)

public static final int HEALTH

Since: API Level 5

常量值: 2304 (0x00000900)

public static final int IMAGING

Since: API Level 5

常量值: 1536 (0x00000600)

public static final int MISC

Since: API Level 5

常量值: 0 (0x00000000)

public static final int NETWORKING

Since: API Level 5

常量值: 768 (0x00000300)

public static final int PERIPHERAL

Since: API Level 5

常量值: 1280 (0x00000500)

public static final int PHONE

Since: API Level 5

常量值: 512 (0x00000200)

public static final int TOY

Since: API Level 5

常量值: 2048 (0x00000800)

public static final int UNCATEGORIZED

Since: API Level 5

常量值: 7936 (0x00001f00)

public static final int WEARABLE

常量值: 1792 (0x00000700)

Public Constructors

public BluetoothClass.Device.Major ()

public static final class

BluetoothClass.Service

extends Object

java.lang.Object

↳ android.bluetooth.BluetoothClass.Service

Class Overview

定义所有的服务类常量

任意 BluetoothClass 由 0 或多个服务类编码。

Summary

常量

int AUDIO

int CAPTURE

int INFORMATION

int LIMITED_DISCOVERABILITY

int NETWORKING

int OBJECT_TRANSFER

int POSITIONING

int RENDER

int TELEPHONY

公共构造器

BluetoothClass.Service()

继承方法

From class java.lang.Object

Object

clone()

创建并返回该对象的复制品.

boolean equals(Object o)

比较带有特定目标的常量。如果他们相等则标示出来。

void finalize()

对象的内存被虚拟机收回前需要调用该方法

final Class<? extends Object>

getClass()

返回一个唯一的 Class 常量，该常量代表这个对象类

int hashCode()

返回该对象的一个整型哈希码

final void notify()

产生一个在该对象监测器中等待被唤醒（通过 wait() 方法进行调用）的线程

final void notifyAll()

产生在对象监测器中等待被唤醒（通过 wait() 方法进行调用）的所有线程

String

toString()

返回这个对象的字符串，该字符串包含精确且可读的介绍

final void wait(long millis, int nanos)

产生一个调用线程，该线程将等待，直到另一个线程调用了 notify()或 notifyAll()方法，或者直到产生一个精确的超时。

final void wait(long millis)

产生一个调用线程，该线程将等待，直到另一个线程调用了 notify()或 notifyAll()方法，或者直到产生一个精确的超时。

final void wait()

产生一个调用线程，该线程将等待，直到另一个线程调用了在该对象中的 notify()或 notifyAll()方法。

Constants

public static final int AUDIO

Since: API Level 5

常量值: 2097152 (0x00200000)

public static final int CAPTURE

Since: API Level 5

常量值: 524288 (0x00080000)

public static final int INFORMATION

Since: API Level 5

常量值: 8388608 (0x00800000)

public static final int LIMITED_DISCOVERABILITY

Since: API Level 5

常量值: 8192 (0x00002000)

public static final int NETWORKING

Since: API Level 5

常量值: 131072 (0x00020000)

public static final int OBJECT_TRANSFER

Since: API Level 5

常量值: 1048576 (0x00100000)

public static final int POSITIONING

Since: API Level 5

常量值: 65536 (0x00010000)

public static final int RENDER

Since: API Level 5

常量值: 262144 (0x00040000)

public static final int TELEPHONY

Since: API Level 5

常量值: 4194304 (0x00400000)

Public Constructors

public BluetoothClass.Service ()

public final class

BluetoothDevice

extends Object

implements Parcelable

java.lang.Object

↳ android.bluetooth.BluetoothDevice

Class Overview

代表一个远程蓝牙设备。让你创建一个带有各自设备的 BluetoothDevice 或者查询其皆如名称、地址、类和连接状态等信息。

对于蓝牙硬件地址而言，这个类仅仅是一个瘦包装器。这个类的对象是不可改变的。这个类上的操作会使用这个用来创建 BluetoothDevice 类的 BluetoothAdapter 类执行在远程蓝牙硬件上。

为了获得 BluetoothDevice,类，使用 BluetoothAdapter.getRemoteDevice(String)方法去创建一个表示已知 MAC 地址的设备（用户可以通过带有 BluetoothAdapter 类来完成对设备的查找）或者从一个通过 BluetoothAdapter.getBondedDevices()得到返回值的有联系的设备集合来得到该设备。

NOTE:需要 BLUETOOTH 类的许可

Summary

常量

String

ACTION_ACL_CONNECTED

广播活动：指明一个与远程设备建立的低级别（ACL）连接

String

ACTION_ACL_DISCONNECTED

广播活动：指明一个来自于远程设备的低级别（ACL）连接的断开

String

ACTION_ACL_DISCONNECT_REQUESTED

广播活动：指明一个为远程设备提出的低级别（ACL）的断开连接请求，并即将断开连接。

String

ACTION_BOND_STATE_CHANGED

广播活动：指明一个远程设备的连接状态的改变

String

ACTION_CLASS_CHANGED

广播活动：一个已经改变的远程设备的蓝牙类。

String

ACTION_FOUND

广播活动：发现远程设备

String

ACTION_NAME_CHANGED

广播活动：指明一个远程设备的昵称第一次找到，或者自从最后一次找到该昵称开始已经改变。

int BOND_BONDED

指明远程设备已经匹配

int BOND_BONDING

指明和远程设备的匹配正在进行中

int BOND_NONE

指明远程设备并未匹配

Creator<BluetoothDevice>

CREATOR

int ERROR

该类的错误标志值

String

EXTRA_BOND_STATE

作为一个 ACTION_BOND_STATE_CHANGED 的整型附加域

String

EXTRA_CLASS

作为一个 ACTION_FOUND and 和 ACTION_CLASS_CHANGED 的 Parcelabe BluetoothClass 附加域。

String

EXTRA_DEVICE

每次通过该类进行广播时，作为 Parcelable BluetoothDevice 的附加域

String

EXTRA_NAME

作为 ACTION_NAME_CHANGED 和 ACTION_FOUND 的字符串附加域

String

EXTRA_PREVIOUS_BOND_STATE

作为 ACTION_BOND_STATE_CHANGED 的整型附加域

String

EXTRA_RSSI

作为 ACTION_FOUND 的可选短整型附加域

继承常量

来自接口 android.os.Parcelable

int CONTENTS_FILE_DESCRIPTOR

使用 describeContents()方法的位屏蔽：每个位代表一种在数据流序列中被认为具有特殊意义的对象

int PARCELABLE_WRITE_RETURN_VALUE

使用 writeToParcel(Parcel, int)方法的标志：这个被写入的对象是一个返回值，它是一个皆如"Parcelable someFunction()", "void someFunction(out Parcelable)", or "void someFunction(inout Parcelable)"等函数的结果。

Public Methods

BluetoothSocket

createRfcommSocketToServiceRecord(UUID uuid)

创建一个 RFCOMM 以准备开始一个对使用 uuid 的 SDP 查找的远程设备进行安全而友好的连接。

int describeContents()

描述了包含在 Parcelable's marshalled representation 中的特殊对象的种类。

boolean equals(Object o)

比较带有特定目标的常量。如果他们相等则标示出来。

String

getAddress()

返回该蓝牙设备的硬件地址

BluetoothClass

getBluetoothClass()

获得远程设备的蓝牙类

int getBondState()

获得远程设备的连接状态

String

getName()

获得远程设备的蓝颜昵称

int hashCode()

返回该对象的一个整型哈希值

String

toString()

返回该蓝牙设备的字符串表达式

void writeToParcel(Parcel out, int flags)

将类的数据写入外部提供的 Parcel 中

继承方法

From class java.lang.Object

Object

clone()

创建并返回该对象的复制品.

boolean equals(Object o)

比较带有特定目标的常量。如果他们相等则标示出来。

void finalize()

对象的内存被虚拟机收回前需要调用该方法

final Class<? extends Object>

getClass()

返回一个唯一的 Class 常量，该常量代表这个对象类

int hashCode()

返回该对象的一个整型哈希码

final void notify()

产生一个在该对象监测器中等待被唤醒（通过 wait() 方法进行调用）的线程

final void notifyAll()

产生在对象监测器中等待被唤醒（通过 wait() 方法进行调用）的所有线程

String

toString()

返回这个对象的字符串，该字符串包含精确且可读的介绍

final void wait(long millis, int nanos)

产生一个调用线程，该线程将等待，直到另一个线程调用了 notify()或 notifyAll()方法，或者直到产生一个精确的超时。

final void wait(long millis)

产生一个调用线程，该线程将等待，直到另一个线程调用了 notify()或 notifyAll()方法，或者直到产生一个精确的超时。

final void wait()

产生一个调用线程，该线程将等待，直到另一个线程调用了在该对象中的 notify()或 notifyAll()方法。

From interface android.os.Parcelable

abstract int describeContents()

描述了包含在 Parcelable's marshalled representation 中的特殊对象的种类。

abstract void writeToParcel(Parcel dest, int flags)

将类的数据写入外部提供的 Parcel 中

Constants

public static final String ACTION_ACL_CONNECTED

Since: API Level 5

指明一个与远程设备建立的低级别（ACL）连接

总是包含 EXTRA_DEVICE.附加域

ACL 连接通过 Android 蓝牙栈自动进行管理

需要 BLUETOOTH 去获取

常量值: "android.bluetooth.device.action.ACL_CONNECTED"

public static final String ACTION_ACL_DISCONNECTED

Since: API Level 5

广播活动：指明一个来自于远程设备的低级别（ACL）连接的断开

总是包含 EXTRA_DEVICE.附加域

ACL 连接通过 Android 蓝牙栈自动进行管理

需要 BLUETOOTH 去获取

常量值: "android.bluetooth.device.action.ACL_DISCONNECTED"

public static final String ACTION_ACL_DISCONNECT_REQUESTED

Since: API Level 5

指明一个为远程设备提出的低级别（ACL）的断开连接请求，并即将断开连接。

对于友好的断开连接，该常量是有作用的。应用程序可以用它作为暗示去马上中断对远程设备的高级别的连接（RFCOMM,L2CAP,或者其它连接）。

总是包含 EXTRA_DEVICE.附加域

需要 BLUETOOTH 去获取

常量值: "android.bluetooth.device.action.ACL_DISCONNECT_REQUESTED"

public static final String ACTION_BOND_STATE_CHANGED

Since: API Level 5

指明一个远程设备的连接状态的改变。

比如，当一个设备已经被匹配。

总是包含 EXTRA_DEVICE, EXTRA_BOND_STATE 和 EXTRA_PREVIOUS_BOND_STATE.这些附加域

需要 BLUETOOTH 去获取

常量值: "android.bluetooth.device.action.BOND_STATE_CHANGED"

public static final String ACTION_CLASS_CHANGED

Since: API Level 5

广播活动：一个已经改变的远程设备的蓝牙类。

总是包含 EXTRA_DEVICE 和 EXTRA_BOND_STATE 这些附加域

需要 BLUETOOTH 去获取

See Also

ERROR(BluetoothClass} /{@link BluetoothClass})

常量值: "android.bluetooth.device.action.CLASS_CHANGED"

public static final String ACTION_FOUND

Since: API Level 5

广播活动：发现远程设备

当一个远程设备在查找过程中被发现时，发送该常量值。

总是包含 EXTRA_DEVICE 和 EXTRA_CLASS.这些附加域。如果可用的话，也可包含 EXTRA_NAME 和/或 EXTRA_RSSI 这些附加域。

需要 BLUETOOTH 去接收

常量值: "android.bluetooth.device.action.FOUND"

public static final String ACTION_NAME_CHANGED

Since: API Level 5

广播活动：指明一个远程设备的昵称第一次找到，或者自从最后一次找到该昵称开始已经改变。

总是包含 EXTRA_DEVICE 和 EXTRA_NAME 这些附加域

需要 BLUETOOTH 去接收

常量值: "android.bluetooth.device.action.NAME_CHANGED"

public static final int BOND_BONDED

Since: API Level 5

指明远程设备已经匹配。

一个共享的连接键为了远程设备而存在于本地，因而设备间的通讯可以被认证和加密。

和远程设备的匹配并不意味着设备间已经成功连接。它只意味着匹配过程已经在稍早之前完成，并且连接键已经存储在本地，准备在下次连接的时候使用。

常量值: 12 (0x0000000c)

public static final int BOND_BONDING

Since: API Level 5

指明和远程设备的匹配正在进行中

常量值: 11 (0x0000000b)

public static final int BOND_NONE

Since: API Level 5

指明远程设备未被匹配。

There is no shared link key with the remote device, so communication (if it is allowed at all) will be unauthenticated and unencrypted.

不存在为了远程设备而已经共享的连接键，因而设备间的通讯（如果完全被允许）不可被认证和加密。

常量值: 10 (0x0000000a)

public static final Creator<BluetoothDevice> CREATOR

Since: API Level 5

public static final int ERROR

Since: API Level 5

该类的错误标志值. 标记该类的错误值。确保和该类中的任意其它整数常量不相等。它为需要一个标记错误值的函数提供了便利。例如： Intent.getIntExtra(BluetoothAdapter.EXTRA_STATE, BluetoothAdapter.ERROR)

常量值: -2147483648 (0x80000000)

public static final String EXTRA_BOND_STATE

Since: API Level 5

作为一个 ACTION_BOND_STATE_CHANGED 的整型附加域。包含了远程设备的匹配状态。

可能值有: BOND_NONE, BOND_BONDING, BOND_BONDED.

常量值: "android.bluetooth.device.extra.BOND_STATE"

public static final String EXTRA_CLASS

Since: API Level 5

作为一个 ACTION_FOUND and 和 ACTION_CLASS_CHANGED 的 Parcelabe BluetoothClass 附加域。

常量值: "android.bluetooth.device.extra.CLASS"

public static final String EXTRA_DEVICE

Since: API Level 5

每次通过该类进行广播时，作为 Parcelable BluetoothDevice 的附加域。它包含了该常量适用的 BluetoothDevice 类。

常量值: "android.bluetooth.device.extra.DEVICE"

public static final String EXTRA_NAME

Since: API Level 5

作为 ACTION_NAME_CHANGED 和 ACTION_FOUND 的字符串附加域。它包含了这个蓝牙昵称。

常量值: "android.bluetooth.device.extra.NAME"

public static final String EXTRA_PREVIOUS_BOND_STATE

Since: API Level 5

作为 ACTION_BOND_STATE_CHANGED 的整型附加域。包含了远程设备以前的匹配状态。

可能值有: BOND_NONE, BOND_BONDING, BOND_BONDED.

常量值: "android.bluetooth.device.extra.PREVIOUS_BOND_STATE"

public static final String EXTRA_RSSI

Since: API Level 5

作为 ACTION_FOUND 的可选短整型附加域。包含了被蓝牙硬件通知的远程设备的 RSSI(Receive Signal Strength Indication，接收信号强度指示)值。

常量值: "android.bluetooth.device.extra.RSSI"

Public Methods

public BluetoothSocket createRfcommSocketToServiceRecord (UUID uuid)

Since: API Level 5

创建一个 RFCOMM 以准备开始一个对使用 uuid 的 SDP 查找的远程设备进行安全而友好的连接。

This is designed to be used with listenUsingRfcommWithServiceRecord(String, UUID) for peer-peer Bluetooth applications.

该方法是为了使用带有 listenUsingRfcommWithServiceRecord(String, UUID)方法来进行对等的蓝牙应用而设计的

使用 connect()初始化这个外界连接。它也将执行一个已给与 UUID 的 SDP 查找，从而确定连接到哪个通道上。

远程设备将被认证，在这个端口上的通讯会被加密。

需要 BLUETOOTH 去接收

参数

uuid 查询 RFCOMM 通道的服务记录 UUID

返回值

一个准备好外界连接的 RFCOMM 蓝牙服务端口

异常抛出

IOException

出现错误，比如蓝牙模块不可用，或者许可无效。

public int describeContents ()

Since: API Level 5

描述了包含在 Parcelable's marshalled representation 中的特殊对象的种类。

Returns

一个指示被 Parcelabel 所排列的特殊对象类型集合的位屏蔽。

public boolean equals (Object o)

Since: API Level 5

比较带有特定目标的常量。如果他们相等则标示出来。 为了保证其相等，o 必须代表相同的对象，该对象作为这个使用类依赖比较的常量。通常约定，该比较既要可移植又需灵活

当且仅当 o 是一个作为接收器（使用==操作符来做比较）的精确相同的对象是，这个对象的实现才返回 true 值。子类通常实现 equals(Object)方法，这样它才会重视这两个对象的类型和状态。

通常约定，对于 equals(Object)和 hashCode() 方法，如果 equals 对于任意两个对象返回真值，那么 hashCode()必须对这些对象返回相同的纸。这意味着对象的子类通常都覆盖或者都不覆盖这两个方法。

参数

o 需要对比常量的对象

返回值

如果特定的对象和该对象相等则返回 true，否则返回 false。

public String getAddress ()

Since: API Level 5

返回该蓝牙设备的硬件地址

例如： "00:11:22:AA:BB:CC".

返回值

字符串类型的蓝牙硬件地址

public BluetoothClass getBluetoothClass ()

Since: API Level 5

获取远程设备的蓝牙类

需要 BLUETOOTH 类

返回值

蓝牙类对象，出错时返回空值、

public int getBondState ()

Since: API Level 5

获取远程设备的连接状态。

连接状态的可能值有: BOND_NONE, BOND_BONDING, BOND_BONDED.

需要 BLUETOOTH 类

返回值

连接状态。

public String getName ()

Since: API Level 5

获取远程设备的蓝牙昵称。

当执行设备扫描的时候，本地适配器将自动寻找远程名称。该方法只返回来自存储器中该设备的名称。

需要 BLUETOOTH 类

返回值

蓝牙昵称，如果出现问题则返回控制。

public int hashCode ()

Since: API Level 5

返回该对象的一个整型哈希值. 通常约定，如果 equals 对于任意两个对象返回真值，那么 hashCode()必须对这些对象返回相同的值。这意味着对象的子类通常都覆盖或者都不覆盖这两个方法。

返回值

该对象的哈希值

public String toString ()

Since: API Level 5

返回该蓝牙设备的字符串表达式。

这是一个蓝牙硬件地址，例如"00:11:22:AA:BB:CC".然而，如果用户明确需要蓝牙硬件地址以防以后 toString()表达式会改变的话，用户总是需要使用 getAddress()方法。

返回值

该蓝牙设备的字符串表达式。

public void writeToParcel (Parcel out, int flags)

Since: API Level 5

将类的数据写入外部提供的 Parcel 中

Parameters

out 对象需要被写入的 Parcel

flags 和对象需要如何被写入有关的附加标志。可能是 0，或者可能是 PARCELABLE_WRITE_RETURN_VALUE。

public final class

BluetoothServerSocket

extends Object

implements Closeable

java.lang.Object

↳ android.bluetooth.BluetoothServerSocket

Class Overview

一个蓝牙监听端口

蓝牙端口监听接口和 TCP 端口类似：Socket 和 ServerSocket 类。在服务器端，使用 BluetoothServerSocket 类来创建一个监听服务端口。当一个连接被 BluetoothServerSocket 所接受，它会返回一个新的 BluetoothSocket 来管理该连接。在客户端，使用一个单独的 BluetoothSocket 类去初始化一个外接连接和管理该连接。

最通常使用的蓝牙端口是 RFCOMM，它是被 Android API 支持的类型。RFCOMM 是一个面向连接，通过蓝牙模块进行的数据流传输方式，它也被称为串行端口规范（Serial Port Profile，SPP）。

为了创建一个对准备好的新来的连接去进行监听 BluetoothServerSocket 类，使用 BluetoothAdapter.listenUsingRfcommWithServiceRecord()方法。然后调用 accept()方法去监听该链接的请求。在连接建立之前，该调用会被阻断，也就是说，它将返回一个 BluetoothSocket 类去管理该连接。每次获得该类之后，如果不再需要接受连接，最好调用在 BluetoothServerSocket 类下的 close()方法。关闭 BluetoothServerSocket 类不会关闭这个已经返回的 BluetoothSocket 类

BluetoothSocket 类线程安全。特别的，close()方法总会马上放弃外界操作并关闭服务器端口。

Note: 需要 BLUETOOTH 类的许可。

Summary

公共方法

BluetoothSocket

accept()

阻塞直到一个连接已经建立

BluetoothSocket

accept(int timeout)

阻塞直到一个带有超时的连接已经建立

void close()

马上关闭端口，并释放所有相关的资源。

继承方法

From class java.lang.Object

Object

clone()

创建并返回该对象的复制品.

boolean equals(Object o)

比较带有特定目标的常量。如果他们相等则标示出来。

void finalize()

对象的内存被虚拟机收回前需要调用该方法

final Class<? extends Object>

getClass()

返回一个唯一的 Class 常量，该常量代表这个对象类

int hashCode()

返回该对象的一个整型哈希码

final void notify()

产生一个在该对象监测器中等待被唤醒（通过 wait() 方法进行调用）的线程

final void notifyAll()

产生在对象监测器中等待被唤醒（通过 wait() 方法进行调用）的所有线程

String

toString()

返回这个对象的字符串，该字符串包含精确且可读的介绍

final void wait(long millis, int nanos)

产生一个调用线程，该线程将等待，直到另一个线程调用了 notify()或 notifyAll()方法，或者直到产生一个精确的超时。

final void wait(long millis)

产生一个调用线程，该线程将等待，直到另一个线程调用了 notify()或 notifyAll()方法，或者直到产生一个精确的超时。

final void wait()

产生一个调用线程，该线程将等待，直到另一个线程调用了在该对象中的 notify()或 notifyAll()方法。

From interface java.io.Closeable

abstract void close()

关闭对象并且释放任意该对象持有的系统资源。

Public Methods

public BluetoothSocket accept ()

Since: API Level 5

阻塞直到一个连接已经建立

在一个成功建立的连接上返回一个已连接的 BluetoothSocket 类。

每当该调用返回的时候，它可以在此调用去接收以后新来的连接。

close()方法可以用来放弃从另一线程来的调用。

返回值

已连接的 BluetoothSocket

异常抛出

IOException

出现错误，比如该调用被放弃，或者超时。

public BluetoothSocket accept (int timeout)

Since: API Level 5

阻塞直到一个带超时的连接已经建立

在一个成功建立的连接上返回一个已连接的 BluetoothSocket 类。

每当该调用返回的时候，它可以在此调用去接收以后新来的连接。

close()方法可以用来放弃从另一线程来的调用。

返回值

已连接的 BluetoothSocket

异常抛出

IOException

出现错误，比如该调用被放弃，或者超时。

public void close ()

Since: API Level 5

马上关闭端口，并释放所有相关的资源。

在其他线程的该端口中引起阻塞，从而使系统马上抛出一个 IO 异常。

关闭 BluetoothServerSocket 不会关闭接受自 accept()的任意 BluetoothSocket。

异常抛出

IOException

public final class

BluetoothSocket

extends Object

implements Closeable

java.lang.Object

↳ android.bluetooth.BluetoothSocket

Class Overview

蓝牙端口监听接口和 TCP 端口类似：Socket 和 ServerSocket 类。在服务器端，使用 BluetoothServerSocket 类来创建一个监听服务端口。当一个连接被 BluetoothServerSocket 所接受，它会返回一个新的 BluetoothSocket 来管理该连接。在客户端，使用一个单独的 BluetoothSocket 类去初始化一个外接连接和管理该连接。

最通常使用的蓝牙端口是 RFCOMM，它是被 Android API 支持的类型。RFCOMM 是一个面向连接，通过蓝牙模块进行的数据流传输方式，它也被称为串行端口规范（Serial Port Profile，SPP）。

为了创建一个 BluetoothSocket 去连接到一个已知设备，使用方法 BluetoothDevice.createRfcommSocketToServiceRecord()。然后调用 connect()方法去尝试一个面向远程设备的连接。这个调用将被阻塞指导一个连接已经建立或者该链接失效。

为了创建一个 BluetoothSocket 作为服务端（或者“主机”），查看 BluetoothServerSocket 文档。

每当该端口连接成功，无论它初始化为客户端，或者被接受作为服务器端，通过 getInputStream()和 getOutputStream()来打开 IO 流，从而获得各自的 InputStream 和 OutputStream 对象

BluetoothSocket 类线程安全。特别的，close()方法总会马上放弃外界操作并关闭服务器端口。

Note: 需要 BLUETOOTH 类的许可。

Summary

公共方法

void close()

马上关闭该端口并且释放所有相关的资源。

void connect()

尝试连接到远程设备。

InputStream

getInputStream()

通过连接的端口获得输入数据流

OutputStream

getOutputStream()

通过连接的端口获得输出数据流.

BluetoothDevice

getRemoteDevice()

获得该端口正在连接或者已经连接的远程设备。

继承方法

From class java.lang.Object

Object

clone()

创建并返回该对象的复制品.

boolean equals(Object o)

比较带有特定目标的常量。如果他们相等则标示出来。

void finalize()

对象的内存被虚拟机收回前需要调用该方法

final Class<? extends Object>

getClass()

返回一个唯一的 Class 常量，该常量代表这个对象类

int hashCode()

返回该对象的一个整型哈希码

final void notify()

产生一个在该对象监测器中等待被唤醒（通过 wait() 方法进行调用）的线程

final void notifyAll()

产生在对象监测器中等待被唤醒（通过 wait() 方法进行调用）的所有线程

String

toString()

返回这个对象的字符串，该字符串包含精确且可读的介绍

final void wait(long millis, int nanos)

产生一个调用线程，该线程将等待，直到另一个线程调用了 notify()或 notifyAll()方法，或者直到产生一个精确的超时。

final void wait(long millis)

产生一个调用线程，该线程将等待，直到另一个线程调用了 notify()或 notifyAll()方法，或者直到产生一个精确的超时。

final void wait()

产生一个调用线程，该线程将等待，直到另一个线程调用了在该对象中的 notify()或 notifyAll()方法。

From interface java.io.Closeable

abstract void close()

关闭对象并且释放任意该对象持有的系统资源。

Public Methods

public void close ()

Since: API Level 5

马上关闭该端口并且释放所有相关的资源。

在其它线程的该端口中引起阻塞，从而使系统马上抛出一个 IO 异常。

异常抛出

IOException

public void connect ()

Since: API Level 5

尝试连接到远程设备。

该方法将阻塞，指导一个连接建立或者失效。如果该方法没有返回异常值，则该端口现在已经建立。

当设备查找正在进行的时候，创建对远程蓝牙设备的新连接不可被尝试。在蓝牙适配器上，设备查找是一个重量级过程，并且肯定会降低一个设备的连接。使用 cancelDiscovery()方法去取消一个外界的查询。查询并不由活动所管理，而作为一个系统服务来运行，所以即使它不能直接请求一个查询，应用程序也总会调用 cancelDiscovery()方法。。

close()方法可以用来放弃从另一线程而来的调用。

异常抛出

IOException

on error, for example connection failure

public InputStream getInputStream ()

Since: API Level 5

通过连接的端口获得输入数据流

即使该端口未连接，该输入数据流也会返回。不过在该数据流上的操作将抛出异常，直到相关的连接已经建立。

返回值

输入流

异常抛出

IOException

public OutputStream getOutputStream ()

Since: API Level 5

通过连接的端口获得输出数据流

即使该端口未连接，该输出数据流也会返回。不过在该数据流上的操作将抛出异常，直到相关的连接已经建立。

返回值

输出流

异常抛出

IOException

public BluetoothDevice getRemoteDevice ()

Since: API Level 5

获得该端口正在连接或者已经连接的远程设备。

返回值

远程设备