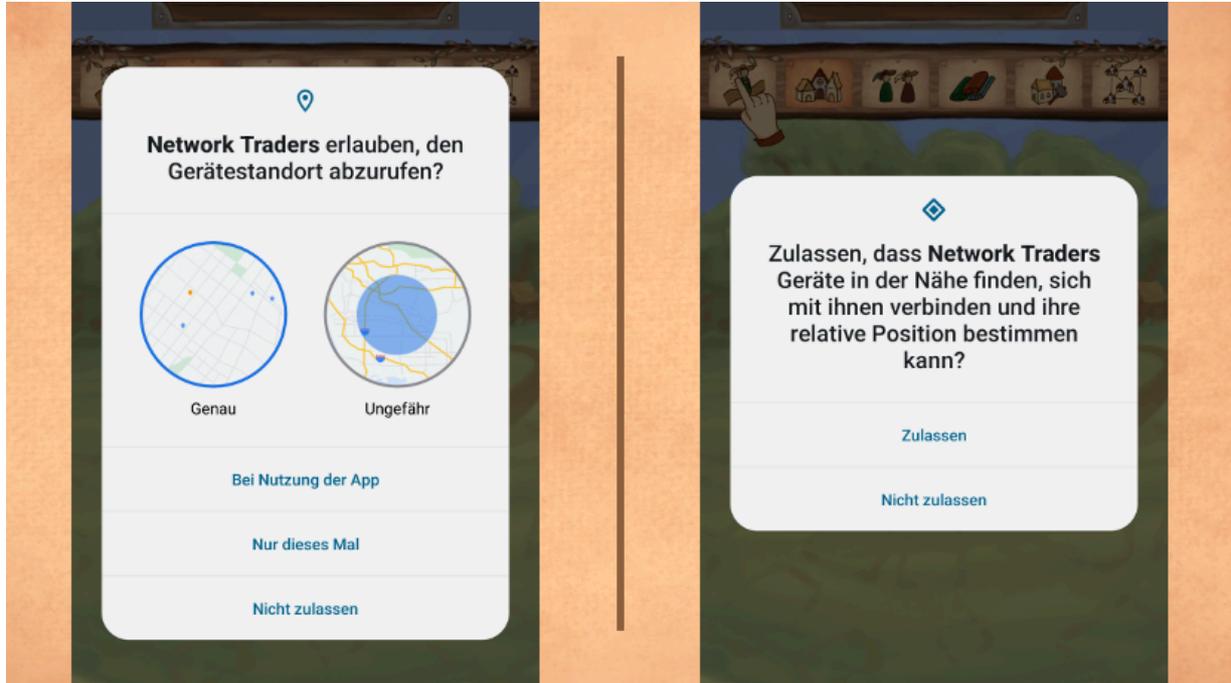


Games:Tech:Blog

A Blog about Games, Technology in Games, and Technology in General



Bluetooth Permissions on Android 12 with Unity (DevBlog #27)

For quite some time now I had problems with the *Nearby Devices* permission necessary for Bluetooth LE from Android 12 upwards. I finally fixed this in Network Traders 0.4.6.1, but not without some trial and error. Here is what I tried and how I finally fixed my permissions issues using the Unity API.

In Network Traders, you send merchants – a kind of independent software agent – from your mobile device to those of other players. A unique feature of the game is that you need to be in the vicinity of the other device, and this is ensured using Bluetooth Low Energy (BLE) signals.

However, BLE is also used for so-called beacons, devices of which the physical location can be tracked via these radio signals. So when playing the game, your mobile device exhibits the same properties as such a beacon. In principle, it is therefore possible to track its location. This may not be desired by the user, so the necessary permissions, such as ACCESS_FINE_LOCATION, are rated as “dangerous” by Android.

Up until Android 11 ([API level 30](#)) it was enough to declare the necessary permissions in your Android manifest. Even the location permissions were not required. The manifest was as simple as this:

```
<uses-permission android:name="android.permission.BLUETOOTH_ADMIN" />
<uses-permission android:name="android.permission.BLUETOOTH" />
```

From API level 31 on, Android got more restrictive. From this release on the Bluetooth permissions were not enough. It became necessary to request the [ACCESS_COARSE_LOCATION](#) and [ACCESS_FINE_LOCATION](#) permissions as well. And even the Bluetooth permissions themselves changed. The [respective Android manifest](#) now looks like this:

```
<uses-permission android:name="android.permission.BLUETOOTH_ADMIN"
android:maxSdkVersion="30" />
<uses-permission android:name="android.permission.BLUETOOTH"
android:maxSdkVersion="30" />
<uses-permission android:name="android.permission.BLUETOOTH_SCAN"
android:usesPermissionFlags="neverForLocation" />
<uses-permission android:name="android.permission.BLUETOOTH_ADVERTISE" />
<uses-permission android:name="android.permission.BLUETOOTH_CONNECT" />
<uses-permission android:name="android.permission.ACCESS_COARSE_LOCATION" />
<uses-permission android:name="android.permission.ACCESS_FINE_LOCATION" />
```

Request Permissions in Java and Unity

However, stating the permissions in the manifest is not enough. You need to request them when you actually want to use the respective functionality. In my case, advertise and receive data over BLE. If you are using your own library written in Java, you can either read the [Android documentation](#) describing the process in detail. Or you can follow the lint suggestions of Android Studio and use code such as this:

```
if (ContextCompat.checkSelfPermission(UnityPlayer.currentActivity,
Manifest.permission.BLUETOOTH_CONNECT) != PackageManager.PERMISSION_GRANTED) {
    if (Build.VERSION.SDK_INT >= Build.VERSION_CODES.S) {
        ActivityCompat.requestPermissions(UnityPlayer.currentActivity, new String[]
{Manifest.permission.BLUETOOTH_CONNECT}, 2);
        return;
    }
}
Intent enableIntent = new Intent(BluetoothAdapter.ACTION_REQUEST_ENABLE);
activity.startActivityForResult(enableIntent, REQUEST_ENABLE_BT);
```

In theory, this should work, but in case you are using a Unity Asset Store package like [Bluetooth Networking](#) you probably do not have access to the underlying code. Fortunately, Unity provides the [Permission](#) class (since Unity 2018.3) which can handle this. The method [Permission.RequestUserPermissions](#), available since Unity

2020.2, allows you to request multiple permissions at once. I recommend using this instead of requesting one permission at a time, which did not work for me. So your Unity script for creating a BLE connection might contain a request like this:

```
#if UNITY_2020_2_OR_NEWER
#if UNITY_ANDROID
if (!Permission.HasUserAuthorizedPermission(Permission.CoarseLocation)
    || !Permission.HasUserAuthorizedPermission(Permission.FineLocation)
    || !Permission.HasUserAuthorizedPermission("android.permission.BLUETOOTH_SCAN")
    ||
!Permission.HasUserAuthorizedPermission("android.permission.BLUETOOTH_ADVERTISE")
    ||
!Permission.HasUserAuthorizedPermission("android.permission.BLUETOOTH_CONNECT"))
    Permission.RequestUserPermissions(new string[] {
        Permission.CoarseLocation,
        Permission.FineLocation,
        "android.permission.BLUETOOTH_SCAN",
        "android.permission.BLUETOOTH_ADVERTISE",
        "android.permission.BLUETOOTH_CONNECT"
    });
#endif
#endif
```

Now, since Network Traders 0.4.6.1, new players should not have to access their settings anymore to grant the “devices nearby” permission manually. At least on my test device, this is not the case anymore. How about yours? I’d really appreciate it if you downloaded the game from [Google Play Store](#) and tried it. I’m looking forward to your feedback!

📅 12. January 2023 👤 Ingo 📁 Android, DevBlog, Game development, Unity 🔖 Android 12, Bluetooth Low Energy, DevBlog, Java, Mobile Game, Nearby Devices, Network Traders, Permissions, Unity

5

Article Rating



✉ Abonnieren ▼

Anmelden

Please login to comment

0 COMMENTS

Data Protection (DE) / Proudly powered by WordPress

