

React Native 0.63.3 Android Build Issues Documentation

Project: zans2-main **Date:** 2025-11-08 **React Native Version:** 0.63.3 **Gradle Version:** 6.2 **Android Gradle Plugin:** 3.5.3

Executive Summary

This document details all issues encountered and attempted fixes while trying to build an old, unmaintained React Native 0.63.3 Android project. The fundamental problem stems from incompatibilities between: - Old React Native project dependencies (from 2020) - Modern Android SDK with API 35+ components - JCenter repository shutdown (May 2021) - Jetifier transformation issues - Java version compatibility with JAXB parser

Status: Build unsuccessful after multiple fix attempts.

Issue 1: Java 14 Incompatibility

Error Message

```
java.lang.NoClassDefFoundError: Could not initialize class org.codehaus.groovy.vmplugin.v7.
```

Root Cause

Gradle 6.2 and Android Gradle Plugin 3.5.3 do not support Java 14+.

Attempted Fix

Downgraded Java version from Java 14 to Java 11, then to Java 8.

```
jabba use adopt@1.11.0-11  # First attempt
jabba use adopt@1.8.0-292  # Second attempt
```

Result

Resolved - Build progressed past this error

Issue 2: JCenter Repository Shutdown & Dependency Resolution

Error Messages

```
Could not find com.facebook.fresco:fresco:2.2.0
```

Could not resolve all files for configuration

Root Cause

JCenter repository was shut down in May 2021. Many dependencies are no longer accessible.

Attempted Fix #1: Replace JCenter with Maven Central

File: /mnt/c/Users/rafeeq.weideman/zans2-main/zans2-main/android/build.gradle

Lines Modified: 10-13, 33-34

Original Code:

```
repositories {  
    google()  
    jcenter()  
}
```

Changed To:

```
repositories {  
    google()  
    mavenCentral()  
}
```

Attempted Fix #2: Force Fresco Version to 2.5.0

File: /mnt/c/Users/rafeeq.weideman/zans2-main/zans2-main/android/app/build.gradle

Lines Added: 184-197

Code Added:

```
configurations.all {  
    resolutionStrategy {  
        force "com.facebook.fresco:fresco:2.5.0"  
        force "com.facebook.fresco:imagepipeline:2.5.0"  
        force "com.facebook.fresco:imagepipeline-native:2.5.0"  
        force "com.facebook.fresco:memory-type-native:2.5.0"  
        force "com.facebook.fresco:memory-type-java:2.5.0"  
        force "com.facebook.fresco:fbcore:2.5.0"  
        force "com.facebook.fresco:imagepipeline-okhttp3:2.5.0"  
        force "com.facebook.fresco:stetho:2.5.0"  
        force "com.facebook.fresco:flipper:2.5.0"  
        force "com.facebook.fresco:memory-type-ashmem:2.5.0"  
    }  
}
```

Attempted Fix #3: Disable Flipper Dependencies

File: /mnt/c/Users/rafeeq.weideman/zans2-main/zans2-main/android/app/build.gradle

Lines Modified: 206-219

Original Code:

```
debugImplementation("com.facebook.flipper:flipper:${FLIPPER_VERSION}") {  
    exclude group: 'com.facebook.fbjni'  
}
```

Changed To:

```
// Flipper dependencies disabled due to Fresco 2.2.0 unavailability in Maven Central  
// Uncomment and update versions if needed for debugging  
// debugImplementation("com.facebook.flipper:flipper:${FLIPPER_VERSION}") {  
//     exclude group: 'com.facebook.fbjni'  
// }
```

Result

Partially resolved - Dependency resolution improved, but new issues emerged

Issue 3: react-native-camera Variant Ambiguity

Error Message

The consumer was configured to find an API of a component,
as well as attribute 'com.android.build.api.attributes.BuildTypeAttr' with value 'debug'.
However we cannot choose between the following variants of project ':react-native-camera':

- generalDebugRuntimeElements
- mlkitDebugRuntimeElements

Root Cause

react-native-camera has multiple product flavors (general, mlkit) and Gradle cannot determine which to use.

Attempted Fix

File: /mnt/c/Users/rafeeq.weideman/zans2-main/zans2-main/android/app/build.gradle

Line Added: 137

Code Added:

```
defaultConfig {  
    applicationId "com.zansshootingsimulator"  
    minSdkVersion rootProject.ext.minSdkVersion
```

```

        targetSdkVersion rootProject.ext.targetSdkVersion
        versionCode 1
        versionName "1.0"
        missingDimensionStrategy 'react-native-camera', 'general' // Added this line
    }

```

Result

Resolved - Variant selection issue fixed

Issue 4: react-native-bluetooth-serial Compilation Error

Error Message

```

RCTBluetoothSerialPackage.java:
Method does not override method from its superclass
@Override
public List<Class<? extends JavaScriptModule>> createJSModules()

```

Root Cause

The `createJSModules()` method was deprecated and removed in React Native 0.47+, but the library still had it.

Attempted Fix

File: /mnt/c/Users/rafeeq.weideman/zans2-main/zans2-main/node_modules/react-native-bluetooth-

Lines Modified: 23-27

Original Code:

```

@Override
public List<Class<? extends JavaScriptModule>> createJSModules() {
    return Collections.emptyList();
}

```

Changed To:

```

// Removed deprecated createJSModules() method (removed in RN 0.47+)
// @Override
// public List<Class<? extends JavaScriptModule>> createJSModules() {
//     return Collections.emptyList();
// }

```

Result

Resolved - Compilation error fixed

Error Message

Root Cause

Attempted Fix #1: Force annotation-experimental Version

Lines Added: 38-42

```
allprojects {
    // ... existing code ...

    configurations.all {
        resolutionStrategy {
            force 'androidx.annotation:annotation-experimental:1.3.0'
        }
    }
}
```

Attempted Fix #2: Exclude annotation-experimental from billing client

Lines Modified: 57-59

```
implementation('com.android.billingclient:billing:3.0.0')
```

Changed To:

```
implementation('com.android.billingclient:billing:3.0.0') {  
    exclude group: 'androidx.annotation', module: 'annotation-experimental'  
}
```

Result: Failed - Same error persisted

Attempted Fix #3: Disable Jetifier

File: /mnt/c/Users/rafeeq.weideman/zans2-main/zans2-main/android/gradle.properties

Line Modified: 26

Original Code:

```
# DISABLED: Jetifier causes build failures with annotation-experimental  
android.enableJetifier=false
```

Attempted Change:

```
android.enableJetifier=false
```

Result: Failed - Jetifier still ran (React Native CLI forces it)

Attempted Fix #4: Clear All Gradle Caches

Commands Executed:

```
rm -rf ~/.gradle/caches/  
rm -rf ~/.gradle/daemon/  
rm -rf android/.gradle/  
rm -rf android/build/  
rm -rf android/app/build/
```

Result: Failed - Same error persisted after cache clear

Final Status

UNRESOLVED - This issue persisted through all fix attempts and appears to be a fundamental incompatibility.

Issue 6: JAXB Parser Errors with Newer Android SDK

Error Messages

Not a number: 35x

java.lang.NumberFormatException: Not a number: 35x

at com.sun.xml.bind.DatatypeConverterImpl._parseInt(DatatypeConverterImpl.java:124)

Not a number: 36.1

unexpected element (uri:"", local:"extension-level"). Expected elements are <{}codename>,<{}extension-level>

unexpected element (uri:"", local:"base-extension"). Expected elements are <{}codename>,<{}]

Root Cause

The Android SDK contains newer package descriptors (API 35, 36) with extension levels that Java 8's JAXB parser cannot parse. These newer SDK formats use "35x" notation for extension levels that the old JAXB implementation doesn't recognize.

Attempted Fix #1: Switch to Java 8

Command Executed:

```
jabba use adopt@1.8.0-292
```

Result: Failed - Java 8's JAXB has same parsing limitations

Attempted Fix #2: Fix Android SDK Path for WSL

File: /mnt/c/Users/rafeeq.weideman/zans2-main/zans2-main/android/local.properties

Line Modified: 1

Original Code:

```
sdk.dir=C:\\Users\\rafeeq.weideman\\AppData\\Local\\Android\\Sdk
```

Changed To:

```
sdk.dir=/mnt/c/Users/rafeeq.weideman/AppData/Local/Android/Sdk
```

Result: SDK path fixed, but JAXB parser errors persisted

Final Status

UNRESOLVED - The old build tools cannot parse newer Android SDK descriptors.

Issue 7: bootClasspath Access in react-native-reanimated

Error Message

Not a number: 35x:

```
java.lang.NumberFormatException: Not a number: 35x
```

```
    at com.sun.xml.bind.DatatypeConverterImpl._parseInt
```

```
    ...
```

```
    at com.android.build.gradle.BaseExtension.getBootClasspath(BaseExtension.java:904)
```

```
    at build_9omz4ynfov1mgmz1gslyvdfol$_run_closure4.doCall(react-native-reanimated/android,
```

Root Cause

react-native-reanimated's build.gradle has Javadoc tasks that access android.bootClasspath, which triggers Android SDK parsing, exposing the JAXB parser issue.

Attempted Fix #1: Comment Out Javadoc Tasks

File: /mnt/c/Users/rafeeq.weideman/zans2-main/zans2-main/node_modules/react-native-reanimated

Lines Modified: 87-98

Original Code:

```
task androidJavadoc(type: Javadoc) {
    source = android.sourceSets.main.java.srcDirs
    classpath += files(android.bootClasspath)
    classpath += files(project.getConfigurations().getByName('compile').asList())
    include '**/*.java'
}

task androidJavadocJar(type: Jar, dependsOn: androidJavadoc) {
    classifier = 'javadoc'
    from androidJavadoc.destinationDir
}
```

Changed To:

```
// Commented out to avoid bootClasspath access issues with newer Android SDK
// task androidJavadoc(type: Javadoc) {
//     source = android.sourceSets.main.java.srcDirs
//     classpath += files(android.bootClasspath)
//     classpath += files(project.getConfigurations().getByName('compile').asList())
//     include '**/*.java'
// }

// task androidJavadocJar(type: Jar, dependsOn: androidJavadoc) {
//     classifier = 'javadoc'
//     from androidJavadoc.destinationDir
// }
```

Result: Progressed past bootClasspath error

Attempted Fix #2: Remove Reference to androidJavadocJar

File: Same file as above

Line Modified: 122

Status: User interrupted before completion

Original Code:

```
artifacts {  
    archives androidSourcesJar  
    archives androidJavadocJar  
}
```

Intended Change:

```
artifacts {  
    archives androidSourcesJar  
    // archives androidJavadocJar // Commented out due to bootClasspath issues  
}
```

Final Status

PARTIALLY RESOLVED - Progressed past bootClasspath error but encountered artifact reference error

Summary of All File Changes

1. android/build.gradle

- **Changes:** Replaced jcenter() with mavenCentral() in two locations, added annotation-experimental force
- **Lines:** 10-13, 33-34, 38-42
- **Status:** Applied

2. android/app/build.gradle

- **Changes:** Added camera variant strategy, forced Fresco versions, disabled Flipper
- **Lines:** 137, 184-197, 206-219
- **Status:** Applied

3. android/gradle.properties

- **Changes:** Jetifier setting toggled multiple times
- **Line:** 26
- **Status:** Currently set to false

4. android/local.properties

- **Changes:** Fixed SDK path for WSL
- **Line:** 1
- **Status:** Applied

5. `node_modules/react-native-bluetooth-serial/.../RCTBluetoothSerialPackage.java`

- **Changes:** Removed deprecated `createJSModules()` method
- **Lines:** 23-27
- **Status:** Applied

6. `node_modules/react-native-iap/android/build.gradle`

- **Changes:** Excluded `annotation-experimental` from billing client
- **Lines:** 57-59
- **Status:** Applied

7. `node_modules/react-native-reanimated/android/build.gradle`

- **Changes:** Commented out Javadoc tasks
 - **Lines:** 87-98
 - **Status:** Applied
-

Root Cause Analysis

The build failures stem from three fundamental incompatibilities:

1. **Legacy Project Architecture:** React Native 0.63.3 from 2020 with unmaintained dependencies
2. **Modern Android SDK:** API 35+ components with extension-level descriptors that old JAXB parsers cannot handle
3. **Repository Migration:** JCenter shutdown forcing migration to Maven Central, but not all dependencies are available

The Jetifier Problem

Jetifier is designed to convert old Android Support Library dependencies to AndroidX. However, it's encountering a library (`annotation-experimental`) that is already in AndroidX format, causing a transformation failure. The tool cannot be disabled because React Native CLI enforces it.

The JAXB Problem

Java 8 and 11's built-in JAXB implementation cannot parse Android SDK package descriptors that use the newer "extension-level" and "base-extension" XML elements introduced in API 33+. The descriptors also use notation like "35x" for extension levels that the parser interprets as invalid numbers.

Recommended Solutions (Not Implemented)

1. **Downgrade Android SDK:** Remove API 35 and 36 packages from the Android SDK, keeping only API 29 and below
 2. **Upgrade React Native:** Migrate to React Native 0.70+ which has better AndroidX and modern SDK support
 3. **Use Older Build Machine:** Set up a build environment with older Android SDK (API 29), Java 8, and cached dependencies
 4. **Fork and Patch Dependencies:** Create local forks of problematic libraries with necessary fixes
 5. **Custom Jetifier Rules:** Create custom Jetifier configuration to skip annotation-experimental transformation
-

Commands Executed

```
# Java version switches
jabba use adopt@1.14.0-2
jabba use adopt@1.11.0-11
jabba use adopt@1.8.0-292

# Build attempts
cd /mnt/c/Users/rafeeq.weideman/zans2-main/zans2-main/android
./gradlew clean
./gradlew clean --no-daemon
npx react-native run-android

# Cache clearing attempts
rm -rf ~/.gradle/caches/
rm -rf ~/.gradle/daemon/
rm -rf android/.gradle/
rm -rf android/build/
rm -rf android/app/build/
rm -rf ~/.gradle/caches/transforms-2/
rm -rf ~/.gradle/caches/transforms-3/
```

Conclusion

After extensive troubleshooting, the build remains unsuccessful due to fundamental incompatibilities between the old React Native 0.63.3 project and modern development tools/SDKs. The issues are:

1. **Jetifier transformation failure** (unresolved)
2. **JAXB parser incompatibility with modern Android SDK** (unresolved)

3. Dependency availability in post-JCenter era (partially resolved)

Recommendation: Either downgrade the entire development environment (Android SDK, build tools) to match the project's era (2020), or invest in upgrading the project to a modern React Native version with current dependencies.