



Mobile User Experience Monitoring - Android

Steps for Bundling eG Mobile Agent with Application

1. Using Downloaded Artefacts

The following steps are required to bundle the 'eG Mobile Agent – Android' into the apps and activate user experience monitoring

1. Download and install mobile agent

- a. Download 'eG Mobile agent' artifacts from eG website and copy to the local drive
- b. Create an 'eg-agent' directory under application's root folder.
- c. Copy agent-gradle-plugin-x.x.x.jar, android-agent-x.x.x.jar and rewriter-agent-x.x.x.jar into this folder just created.

2. Add the following lines into project's build.gradle file:

```
buildscript {  
    repositories {  
        flatDir {  
            dirs 'eg-agent'  
        }  
    }  
    dependencies {  
        classpath fileTree(include: ['*.jar'], dir: 'eg-agent')  
    }  
}
```

3. Add the following lines into application's build.gradle file:

```
repositories {  
    flatDir {  
        dirs '../eg-agent'  
    }  
}  
apply plugin: 'android'  
apply plugin: 'eg'  
dependencies {  
    implementation fileTree(dir: '../eg-agent' , include: ['a*.jar'])  
}
```

4. Set application permissions

Ensure that application requests INTERNET and ACCESS_NETWORK_STATE permissions by adding below lines to application's AndroidManifest.xml under manifest tag. Add only if it's not already present.

```
<uses-permission android:name="android.permission.INTERNET" />  
<uses-permission android:name="android.permission.ACCESS_NETWORK_STATE"/>
```

Also add ***android:usesCleartextTraffic="true"*** in the application tag of the AndroidManifest.xml file as shown below.

```
<application      android:name=".MyApplication"

    android:allowBackup="true"

    android:icon="@mipmap/logo"

    android:label="@string/app_name"

    android:roundIcon="@mipmap/logo"

    android:supportsRtl="true"

    android:theme="@style/AppTheme"

    android:usesCleartextTraffic="true">
```

5. Start the agent

Start the eG Mobile agent while starting the application by implementing the below steps in application's **Main or default activity**.

- a) In application's Main or Default Activity import the eG Mobile Agent class as mentioned in the below box.

```
import com.eg.agent.android.eGAndroidAgent;
```

- b) Copy paste the marked content as shown in the screen shot below which is present in the add/modify component page, into mobile application's onCreate() method of Main or Default Activity.

The screenshot shows a 'Add Component' dialog box with a 'Back' button in the top right corner. The dialog contains a 'Component Information' section with the following fields:

- Category: All
- Component type: Mobile RUM
- Nick name: eGSampleApplication
- Mobile RUM collector: Default Collector (dropdown menu)
- Remote agent: 172.16.14.88 (dropdown menu)
- Application type: ☒ Native ☐ Hybrid

An 'Add' button is located at the bottom center of the dialog.

Android

Manual Android Studio Plugin Maven

01 Download the PDF and follow the steps to inject eG Android Agent into your mobile apps [Download](#)

02 Download eG Android Agent [Download](#)

03 Include this line into Application's Main or Default activity to inject eG Android Agent into your mobile apps [Copy to Clipboard](#)

```
eGAndroidAgent.withCollectorHost("c1fa936b-c0dd-4c50-828e-8a84acc3b861-1645787480197","http://172.16.14.88:7077").start(getApplicationContext());
```

Note:- Starting the eG mobile agent in any other class is not supported and can cause unexpected or unstable behaviour.

6. Build and Run the application

Clean the project and Build the application. While building the application, the console should log messages about transforming classes, artefacts etc. After successful build, run the application in an emulator or mobile device and login to eG manager application to start seeing data.

2. Using Artefacts from Maven Repo

The following steps are required to bundle the 'eG Mobile Agent – Android' into the apps and activate user experience monitoring

1. Add the following bolded lines into project's build.gradle file only if its not present already:

```

buildscript {
    ....
    repositories {
        google()
        jcenter()
        mavenCentral()
    }

    dependencies {
        ....
        classpath "com.eginnovations.agent.android:agent-gradle-plugin:2.0.0"
    }
}

```

2. For below Gradle 7 version:

i. Add the following lines into application's build.gradle file:

```

apply plugin: 'com.android.application'
apply plugin: 'eg'
.....
repositories {
    mavenCentral()
}
.....
dependencies {
    ....
    implementation "com.eginnovations.agent.android:android-agent:2.0.0"
}

```

For Gradle 7 and higher version:

i. Add the following lines into application's build.gradle file:

```

apply plugin: 'com.android.application'
apply plugin: 'eg'
.....
.....
dependencies {
    ....
    implementation "com.eginnovations.agent.android:android-agent:2.0.0"
}

```

ii. Add the following line into project's settings.gradle file:

```

dependencyResolutionManagement {
    repositoriesMode.set(RepositoriesMode.FAIL_ON_PROJECT_REPOS)
    repositories {
        google()
        mavenCentral()
    }
}

```

3. Set application permissions

Ensure that application requests INTERNET and ACCESS_NETWORK_STATE permissions by adding below lines to application's AndroidManifest.xml under manifest tag. Add only if it's not already present.

```
<uses-permission android:name="android.permission.INTERNET" />
<uses-permission android:name="android.permission.ACCESS_NETWORK_STATE"/>
```

Also add ***android:usesCleartextTraffic="true"*** in the application tag of the AndroidManifest.xml file as shown below.

```
<application      android:name=".MyApplication"

    android:allowBackup="true"

    android:icon="@mipmap/logo"

    android:label="@string/app_name"

    android:roundIcon="@mipmap/logo"

    android:supportsRtl="true"

    android:theme="@style/AppTheme"

    android:usesCleartextTraffic="true">
```

4. Start the agent

Start the eG Mobile agent while starting the application by implementing the below steps in application's **Main or default activity**.

- a. In application's Main or Default Activity import the eG Mobile Agent class as mentioned in the below box.

```
import com.eg.agent.android.eGAndroidAgent;
```

- b. Copy paste the marked content as shown in the screen shot below which is present in the add/modify component page, into mobile application's onCreate() method of Main or Default Activity.

Add Component ⓘ

Back

Component Information

Category

All

Component type

Mobile RUM

Nick name

eGSampleApplication

Mobile RUM collector

Default Collector

Remote agent

172.16.14.88

Application type

☒ Native☐ Hybrid

Add

Android

iOS

Manual

Android Studio Plugin

Maven

01

Download the PDF and follow the steps to inject eG Android Agent into your mobile apps

Download

02

Include this line into Application's Main or Default activity to inject eG Android Agent into your mobile apps

Copy to Clipboard

```
eGAndroidAgent.withCollectorHost("c1fa936b-c0dd-4c50-828e-8a84acc3b861-1645787480197","http://172.16.14.88:7077").start(getApplicationContext());
```

Note:- Starting the eG mobile agent in any other class is not supported and can cause unexpected or unstable behaviour.

5. Build and Run the application

Clean the project and Build the application. While building the application, the console should log messages about transforming classes, artefacts etc. After successful build, run the application in an emulator or mobile device and login to eG manager application to start seeing data.

Proguard Configuration for Android Applications

Steps for supporting ProGuard with Eg Agent for Android:

1. Add the following to your ProGuard config file (proguard.cfg, proguardandroid.txt, proguardrules.pro, etc.)
 - a) To support Eg instrumentation, add keep class.
`-keep class com.eg.** { *; }`
 - b) To disable warnings related to unresolved problems, add don't warn.
`-dontwarn com.eg.**`
 - c) To preserve line numbers for crash reporting, add LineNumberTable.
`-keepattributes Exceptions, Signature, InnerClasses, LineNumberTable, SourceFile, EnclosingMethod`
 - d) To preserve stack information for handled exceptions, add SourceFile and EnclosingMethod, as shown above.
2. Clean and build your project.
3. Run your application in an emulator or Mobile device.
4. Login to eG Manager application to start seeing data.