

Version	Date	Description	Resp.
1	29/06-2018	Release doc for CDO version 5.0.x	MOH
2	7/8-2018	Updated language use, introduced summary and minor structure changes	IAN/BJ
3	8/8-2018	Adding a chapter to clean old dependencies	BJ

Summary

This document highlights the key issues to be aware of when upgrading from a CDO version lower than 5.0. There are three main areas described in brief below and in more detail later in the document:

Important API changes

- New parameter to be added to `startCalldorado()`
- GDPR Compliancy, including the rights to be forgotten

Things to be aware of:

- Proguard issue with app icon and name
- New added libraries can conflict
- Prevent multiple instances of the app
- Layout of native field for aftercall
-

New features:

- How to set colors for the new aftercall screen
- Show information of latest phone call

Add `savedInstanceState` as Parameter to `startCalldorado(..)`

We have added the Bundle `savedInstanceState` as a parameter to all overloaded versions of `Calldorado.startCalldorado(..)`. Press 'ctrl' and hover mouse over your version of `startCalldorado(..)` to get the position of the added parameter and feed the variable `savedInstanceState` from your `Activity.onCreate(..)` as parameter. This change was added to prevent possible bugs related to starting more than one instance of the Calldorado initialization process.

Calldorado Handling of 2018 GDPR Rules for EEA Countries

To make Calldorado compliant with the GDPR rules of the EEA countries, we have made following changes to Calldorado:

1. During the first initialization of Calldorado, a load timer will be shown until the screen displays opt-in/permissions for EEA users.
2. We have removed the option to share information about a phone number if the number origin is an EEA country.

3. We have added two new entries to Calldorado settings under the heading 'Data'. Here a user from EEA countries will get options to either enable/disable personalized ads or to erase memory of the app.

Activating the GDPR rules in Calldorado is the new default, but can be deactivated through the following entry in your AndroidManifest under the <Application> tag:

```
<meta-data
    android:name="com.callorado.EEAMode"
    android:value="disable" />
```

Trigger Actions on User Resetting Personal Data

In relation to GDPR, a user has the rights to clear app history. Due to this requirement, users in EEA countries can select 'Delete App Data' from the Calldorado settings menu. If the delete option is selected, the app will be reset like clearing app data through the device settings. To be fully GDPR compliant at data reset, we trigger the method below, which allows a host app to clear data stored in files or reset firebase data.

```
public static void onCdoDataReset(Context context) {
    //Example. Resetting firebase on CDO data reset
    FirebaseAnalytics mFirebaseAnalytics =
    FirebaseAnalytics.getInstance(context);
    mFirebaseAnalytics.resetAnalyticsData();
}
```

The method above is triggered using reflection. The method must be placed in the launch activity of your app and it must have the exact same name or reflection will throw an exception.

App Icon and Name Not Showing Properly in CDO Opt in

Using Proguard obfuscation can lead to issues in showing the app icon and name in the Calldorado initialization screen properly. The bug can be fixed by adding the following rule to your Proguard file:

```
-keep class c.** { *; }
-dontwarn c.**
```

Calldorado 5.x New References

With Calldorado 5.x we have added following references:

- com.android.support:cardview-v7:26.1.0
- com.android.support:appcompat-v7:26.1.0
- com.android.support:design:26.1.0

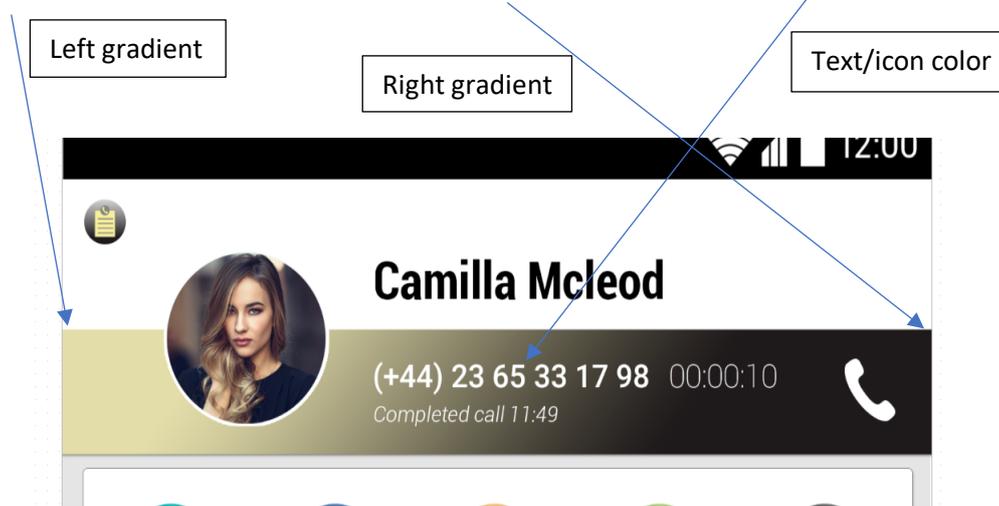
If you have one or more of these references in your local project, you may experience version conflicts. Version conflicts can usually be resolved through changing the local version to the one in the CDO SDK or by excluding the conflicting reference from the CDO import. Excluding one or more CDO references can be triggered as follows but be aware that excluding may lead to issues.

```
implementation ('com.callorado:callorado-release:+@aar') {
    transitive = true
    exclude group: 'com.android.support', module: 'appcompat-v7'
}
```

Aftercall Customized Colors

We have reduced the number of customizable colors on the Aftercall screen to maintain consistency and design standards. The Aftercall screen and in later CDO versions, CDO settings, have one layout element with customizable colors to represent app theme colors. Customized colors can be triggered through:

```
Callorado.setCalloradoAftercallColors(context, Color.parseColor("#FFFFFF"),
Color.parseColor("#e2dca8"), Color.parseColor("#000000"));
```



The above method can be triggered before or after the `Callorado.startCallorado` method in your launcher Activity. To ensure that these theme colors are set immediately after upgrading to CDO 5.0, the `setCalloradoAftercallColors(..)` can be triggered from an upgrade receiver listening for "android.intent.action.MY_PACKAGE_REPLACED". Default colors are white text/icons with gray left gradient and black right gradient.

Native Action on Aftercall

The option to integrate custom content in the Aftercall screen works more or less as before and will in many cases require no changes from you. The native layout will now be wrapped inside a `CardView` to fit the overall layout.

External dependencies

CallDorado relies on external dependencies to serve ads and functionality. If already using dependencies such as Facebook or Google in your project, beware that conflicting versions may cause build issues. You may either align the versions imported or exclude the import locally or from the CDO dependency.

Excluding a dependency from CDO can be performed as follows:

```
dependencies {
    compile('com.calldorado:Calldorado-release:+@aar') {
        transitive = true
        exclude group: 'com.google.android.gms', module: 'play-services-ads'
    }
}
```

If excluding or running a different version of a dependency than the version coming with the CDO version, there is high risk of performance issues even when replacing the dependency with another version.

To ensure that your dependencies gets updated properly when upgrading CDO, make sure to 'clean – refresh-dependencies' after changing the version in the Gradle file.

Prevent Multiple Instances When Navigating from Aftercall to Host App

To improve Google compliancy by linking the host app and CDO closer together, we have added a link from Aftercall screen back to the host app on the top left corner app icon. The link should respect the stack of the host app if already present, but to prevent the possibly of adding launch Activity on top of the existing stack, you will need to add the following entry to the onCreate() of launch Activity (place just after super.onCreate(sav..));

```
if (!isTaskRoot() && getIntent() != null &&
    getIntent().getBooleanExtra("onAppIconClick", false))

    Log.d("test", "finishing activity onCreate()");
    finish();
    return;
}
```

To tell the host app that it has been started or resumed from a CDO Aftercall, we send a Boolean extra with the intent. To set special behavior if the host app is started from CDO, place the following entry in the onCreate(..) of launch Activity:

```
if (getIntent() != null && getIntent().getBooleanExtra("onAppIconClick", false))
{ Your code }
```

New Option to Show Information of Latest Phone Call

CallDorado.showLastCallScreen(context); will show an Aftercall screen containing the information from the latest phone call of a user.