

DDLC Mod Template Guide to Android Mod Making

Works on Ren'Py 6.99.12.4 and 7.3.5

Author: GanstaKingofSA#0235 | u/GanstaKingofSA

For DDLC Mod Template Versions: 2.4.0 and higher

Disclaimer: This guide is not endorsed by Team Salvato. Modding should be done when you have completed DDLC. This is also not meant to port mods without author permission. Do not port other people's mods without their permission or release them on Google Play, Galaxy Store, etc.

Contents

Setting Up The Template	3
Building the Mod for Android	5
Notes	7
FAQ.....	8
Acknowledgements to.....	9

Setting Up The Template

To setup the template on a fresh copy of DDLC to use the Android feature, you'll need the following.

1. A **new install** of DDLC from either <https://ddlc.moe> or from Steam (only tested on the DDLC.moe copy)
2. **DDLC Mod Template Version 2.4.0** or higher
 - a. Please note this is only hosted on <https://github.com/GanstaKingofSA/DDLCModTemplate2.0/releases> , not the Monika After Story Download.
 - b. Extract the template to the folder where **DDLC.exe** [**DDLC.sh** for Linux] is located in and accept any replaces.
 - i. MacOS/OS X users will have to access this via **DDLC.app/Contents/Resources/autorun**
3. **renpy-6.99.12.4-sdk** or **renpy-7.3.5-sdk**
 - a. If you are using Ren'Py 6 (6.99.12.4)
 - i. Download **renpy-6.99.12.4-sdk** by downloading it off <https://renpy.org/release/6.99.12> . Download either **7z exe**, **dmg**, or **tar.bz2** for your respective OS and extract it somewhere.
 - b. If you are using Ren'Py 7 (7.3.5)
 - i. Download **renpy-7.3.5-sdk** by downloading it off <https://renpy.org> and click Download Latest Version. Download either **7z exe**, **dmg**, or **tar.bz2** for your respective OS and extract it somewhere.
4. **Visual Studio Code**
5. **AdoptOpenJDK 8** or **Java JDK 8**
 - a. If you don't have Java JDK 8 installed, install AdoptOpenJDK 8 instead on <https://adoptopenjdk.net/?variant=openjdk8&jvmVariant=hotspot>
6. **RPATool**
 - a. You can obtain RPA Tool on <https://github.com/Shizmob/rpatool/archive/master.zip> . Place the rpatool file inside the master ZIP -> rpatool-master in the game folder.
7. **Python 2.7**
 - a. You can download Python on <https://www.python.org/downloads/release/python-2718/>
8. (Optional) **An android phone** with USB Debugging **On** or **Android Studio** for debugging errors, testing, etc.
 - a. If you are using your own android phone:
 - i. To turn on USB debugging, go to Settings -> About -> Software information and tap build number around 5-8 times repeatedly until it says Developer mode has been enabled. Then go back to Settings -> Developer options and scroll down till you see USB debugging and turn it on.
 - b. If you prefer to use Android Studio instead:

- i. Make sure to have Android Virtual Device checked during setup and install everything. Next, create a project, go to Tools -> AVD Manager and click the play button to start a virtual android phone for debugging use.

NOTE: To access the contents of **DDL.C.app**, right-click DDL.C in MacOS and click **Show Package Contents**.

Installation Steps:

1. Open the mod template ZIP file and drop its contents from the ZIP to the DDL.C base directory where **DDL.C.exe** [DDL.C.sh for Linux] is located. Accept any replaces.
 - a. MacOS/OS X users will have to access this directory via
DDL.C.app/Contents/Resources/autorun
2. Run DDL.C via DDL.C.exe, DDL.C.app, or via the Linux Terminal by typing ./DDL.C.sh
3. The game should run, and you can proceed to make your mod and follow the rest of the guide to setting variables and building your mod for Android.
- 4.

If you are upgrading your template to 2.4.0, please read this section carefully.

Upgrading from MAS Versions 0.2.0 to 1.1.2 – This update like the original 2.X.X updates have major changes to the code to be like DDL.C which MAS' template lacks and especially with the android code for this update. It is highly suggested to move all code you edited to the 2.4.0 files to obtain all DDL.C functionality, your code, and Android functionality in your mod or follow how 2.4.0 is formatted and follow this guide still.

Upgrading from Versions 2.1.X to 2.3.1 – Due to the Android code, a bit of some functionality will require a transfer of code as well to 2.4.0. This changes portions of the *poem game*, *splash*, *script* additions, *options* additions, *definition* additions and *gui* code edits to make DDL.C work on a mobile device. I suggest transferring the code over or follow how 2.4.0 is formatted and follow this guide still.

Building the Mod for Android

Installation Steps:

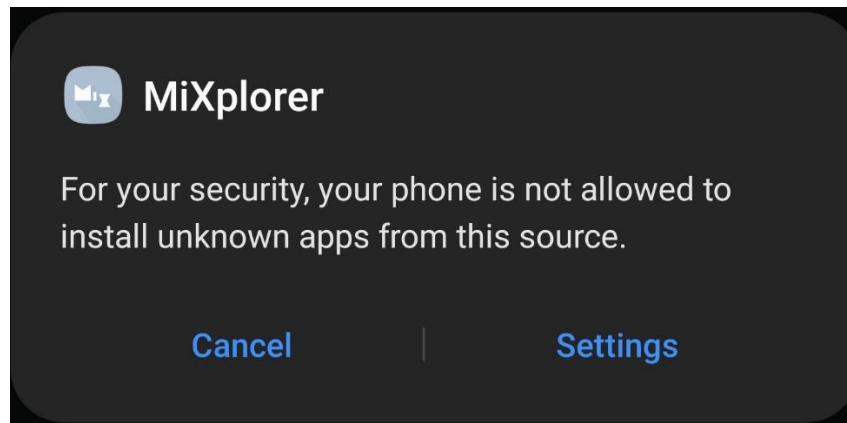
1. Open Terminal, Command Prompt or PowerShell in the game folder and run the following commands
 - a. `python rpatool -x audio.rpa`
 - b. `python rpatool -x fonts.rpa`
 - c. `python rpatool -x images.rpa`
 - i. This part must be done as for now Android does not support RPAs. You must extract DDLC's files to load its files with your mod files. Make sure you BACKUP your mod files before running this else they might get overridden.
 - ii. **Keep the RPA files in the game folder!** At minimum remove the archive check in *splash.rpy* if you must or Ren'Py won't compile your mod on PC or Android since you are technically building on PC. The same goes for making a build for PC/Mac/Linux and Android.
2. Start Ren'Py Launcher and select your mod from the Projects Sidebar, then click **Android**
 - a. You will be asked to download RAPT (Ren'Py Android Platform Tools). Select Yes and it will download RAPT. It will restart Ren'Py Launcher after it installs.
3. Go back to the **Android** section and click **Install SDK and Select Keys**. It will download the Android SDK and other SDK items. Accept the Android Terms and say yes to setting up a key and to backing it up. Set your organization name to your name.
4. Click **Configure** and follow the steps below to setup the mod configuration for Android
 - i. For Full Name, set your mod name
 - ii. For Short Name, set the abbreviation of your mod name
 - iii. For Package Name, use *com.* then your name with a period and the build name of your mod. For example, if your name is Monika and your build name for your mod is MAS in *options.rpy* (under *build.name*), it will be formatted like this. *com.monika.mas*
 - iv. For Application Version, set the version number of your mod.
 - v. For Version Code, leave it as is.
 - vi. Set display, set it to landscape orientation.
 - vii. For app store in-app purchasing, select Neither.
 - viii. Select No to creating an expansion APK.
 - ix. (For 6.99.12.4) For version of Android, select Android 4.0.
 - x. Select No to allowing the mod to access the internet.
 - xi. (For 7.3.5) Select Yes to automatically updating the project.
5. Select **Build Mod** and your mod should compile to a APK to be used on Android. You may then sideload it to your phone and test it for errors or enjoy it on the go. See **Notes** however for some notices that need to be address.
 - a. If you want to test it on a virtual phone or your own phone, make sure either your virtual phone is on, or your phone is connected to your Computer with USB

debugging **on**, select **Build Mod & Install** and launch the app once Ren'Py finishes building and test your mod.

- b. For Ren'Py 6, your mod will be compiled with the following name format. Here is an example with mod name *Cookie Club* and version number *2.3.0*
 - i. *CookieClub-2.3.0-release.apk*
 - c. For Ren'Py 7, your mod will be compiled and output 4 APK files. For universal compatibility between different phone CPU models, use the following name format APK. Here is an example with author name *Natsuki*, mod build name *mlhnstlc* and version number *1.0.0*
 - i. *com.natsuki.mlhnstlc-10000-universal-release.apk*
6. After you tested that your mod works as is and some adjustments if they were needed, ZIP up your APK or upload it to Google Drive, Mega, etc. and your port should be complete.
- a. Make sure to add that the user must turn on Unknown Sources on their phones temporarily and of a Google Play Protect warning that may occur and to tap **Install Anyways**. In addition, Google will ask to send the APK over to them. This does not break the IPG so it's their choice to select Send or Don't Send. In my opinion, just tap Don't Send and you should be good to go. Refer to **Notes** for more details.

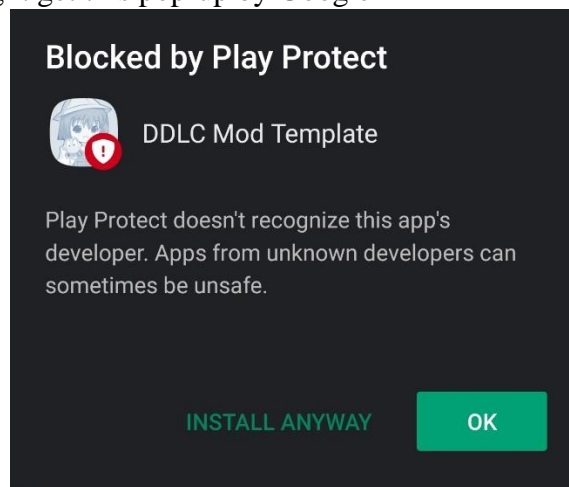
Notes

1. When sideloading your mod either via a file manager or by downloading off Chrome, or any other browser, etc., you may get this pop-up by Android or something similar.



To fix this, tap Settings -> and Turn on Unknown Sources. Android will sometimes allow you to turn this on temporarily for just this one install which is fine. Select either OK or Just Once and tap Install.

2. Additionally, you might get this pop-up by Google



This is Google detecting that you the mod developer are someone they don't recognize and think the app is malicious. Tap Install Anyway and the installation should proceed as normal.

3. While running the game you might get a pop-up to appear that asks whether to send your mod's APK to Google for analysis. This is your choice/player's choice to send it or not. This doesn't violate Team Salvato's IPG as this is a Google Security Message.

FAQ

Q. Where are all the game files/character files located now?

A. They are located in the APK itself however some files can't be loaded from it and are copied over to the *Android/data/com.[yourname].[modbuildname]* for deleting files, adding files, etc.

Q. A bunch of my script is broken. Why is it the case now?

A. Ren'Py on Android runs differently than on PC. Items like *renpy.file* to try a file if it exists or opening it isn't supported in that way and have alternative call names like *file* instead. Refer to the template comments for more information.

Q. Does it have to be *com.[yourname].[modbuildname]* to build the mod? It looks cheezy and bland...

A. No. You can change it to be whatever, but it must follow the Android naming scheme of *x.y.z*

Q. My GUI elements are all broken. What gives here?

A. The GUI loaded is of DDLC and are at the bottom section of *gui.rpy*. Change the variables given there to adjust your GUI to how it is in-game on PC.

Q. Isn't distributing the APK with DDLC files against the IPG?

A. No. Team Salvato acknowledges this is a limitation on Android and allows it however they will keep an eye on how these APK files are distributed.

Q. What is the difference between this and JoiPlay?

A. This runs natively on Android versus an emulation of PC Ren'Py code. This means it is prone to less bugs and failed mods to load than emulation.

Q. How would I add my files that need to be run to the *Android/data/com.[yourname].[modbuildname]* folder like *poemwords.txt*?

A. First run a check to see if the file exists in the area. If it excepts then make Ren'Py write the file in the APK that you want to that folder. Refer to *script-poemgame.rpy* on how that is used. Make sure to add the *if renpy.android:* section in the code. It's very important.

This FAQ is not an exhaustive list of questions for Android DDLC modding development. For further questions, ask in the *#mod_help* channel or *#mod_faq* channel on the DDMC Discord linked here or contact me in DM for any analysis of what might be wrong, though I suggest the DDMC Discord even better for quick help than waiting for me to wake up.

DDMC Discord Invite: <https://discord.gg/Gx7m9xE>

Acknowledgements to

Tom Rothamel i.e. PyTom for Ren'Py and Ren'Py Android functionality

Dan Salvato for making DDLG and allowing us to mod it to our hearts content that is IP compliant.

You for reading this whole guide.

Weiss Schnee for support