

The DDLC Mod Template Guide to Android Mod Making

For Ren'Py 6.99.12.4 and Ren'Py 7.4.5

Guide Version 1.5

Author: GanstaKingofSA#0235 | u/GanstaKingofSA

Android Guidelines

Last Updated on August 20th, 2021

Disclaimer

This guide is not affiliated or endorsed in anyway with Team Salvato (Dan Salvato LLC). Neither Dan Salvato, nor any current or former Team Salvato employees took part in the writing of this guide. This guide is solely made by GanstaKingofSA with the assistance of other people to which are credited in the [Acknowledgements](#) page of this guide.

Mods on Android

This guide is not meant to allow you to port DDLC itself over to Android, nor port the mods made by other mod developers onto the platform without their permission. This guide is meant for you to port your **own** mod project over to the platform or port other mod developers work with their explicit permission.

As per Team Salvato's IPG (Intellectual Property Guidelines), you may not port DDLC itself over to Android nor release it onto the Google Play Store, Amazon Appstore, Samsung's Galaxy Store, Huawei's App Store, Xiaomi's Mi GetApps, Oppo's App Market, Vivo's App Store, FDroid, Aurora Store, Kuan, Aptoide, APKMirror, Itch.io, or any other first-party/third-party Android app store. This also applies to DDLC mods. Mods that are released onto these platforms will be reported to Team Salvato.

As this is meant to port DDLC mods over to Android, as per normal DDLC mods, you may not hold GanstaKingofSA or Team Salvato liable for the mods that are released onto the Android platform and their behaviors. GanstaKingofSA and Team Salvato cannot guarantee that a mod released onto the platform either contains malware, or cause harm to your device or personal files. Installing an Android mod onto your phone is under your discretion.

Everything Else

This disclaimer is not necessarily exhaustive. For questions about what aspects of DDLC may be included in your port, you may contact GanstaKingofSA or Team Salvato for clarification.

Thank you for taking the time in reading this disclaimer. This disclaimer is not meant to discourage people to make their mods available on Android or make an Android mod, rather express the desire to make IP compliant and mod developer permitted Android DDLC-related content. We hope to see mod developers make their mods available on Android and see fans of the game enjoy DDLC mods on-the go and enjoy DDLC mods in many different ways.

Contents

Android Guidelines.....	2
Disclaimer	2
Mods on Android	2
Everything Else	2
Setting Up the Template	4
Installation Steps.....	5
Upgrading Ren'Py/The Template	6
Upgrading from the <i>Monika-After-Story/RationalPi Template</i> (Versions 0.1.0 to 1.1.2)	6
Upgrading from Versions 2.1.X to 2.3.1 (includes 2.3.1-uX versions).....	6
Upgrading from Versions 2.4.0 to 2.4.5 (for templates above 2.4.5)	6
Android by following a <i>DDLC Guide</i>	6
Android under Ren'Py 7.3.5 - 7.4.4 or 7.4.6 and higher.....	6
Building the Mod for Android	7
Notes	9
FAQ.....	11
Acknowledgements	13

Setting Up the Template

To get started, you will need the following.

1. A system with at least 8 GB of RAM.
 - a. Depending on the size of your mod, the RAM amount listed here may increase or decrease in size. Mods which files, along with extracted DDLC assets total more than 500MB will need more than 4 GB RAM in order to build properly.
2. A fresh copy of DDLC from [DDLC.moe](https://doki-doki-literature.club/).
3. DDLC Mod Template Version 2.4.0 or higher from [GanstaKingofSA](#)'s template repository.
 - a. This will **not** work on the Monika-After-Story/RationalPi template or by decompiling the game yourself. For more information, see **Upgrading from the Monika-After-Story/RationalPi Template (Versions 0.2.0 to 1.1.2) and/or Android by following a DDLC Guide**.
4. Ren'Py 6.99.12.4 or Ren'Py 7.4.5 SDK
 - a. Ren'Py versions 7.3.5 – 7.4.4 and 7.4.6+ will not work. For more information, see **Android under Ren'Py 7.3.5 – 7.4.4 or 7.4.6 and higher**.
 - b. If you are using *Ren'Py 6.99.12.4*
 - i. Download the Ren'Py 6.99.12.4 SDK [here](#).
 - c. If you are using *Ren'Py 7.4.5*
 - i. Download the Ren'Py 7.4.5 SDK [here](#).
 - d. Download either the *7z.exe*, *dmg*, or *tar.bz2* package for your respective system and extract it somewhere.
 - e. Make a folder for your mod inside the extracted Ren'Py SDK folder.
5. Visual Studio Code + the Ren'Py Extension from [Microsoft](#).
 - a. To install the Ren'Py Extension, go to *Extensions* (Button with 4 boxes on the left side), search *Ren'Py* and install the first extension in the results bar.
6. Eclipse Temurin 8 from [Adoptium](#).
 - a. Building your mod while having Temurin 8 & 11 installed may cause problems. See **Notes** Section D for more information.
7. RPATool from [Shizmob](#)'s repository.
8. Python from [Python.org](https://python.org).
 - a. Make sure to check *Add python to PATH* when installing Python to your system.
9. An android phone.

Installation Steps

Note: If you are upgrading your template to 2.4.X, followed a guide, or used a tool to make your mod project, please read **Upgrading Ren'Py/The Template** before proceeding.

Note for MacOS Users: To access the directories inside DDLC.app, right-click on it and select *Show Package Contents*.

1. Open the *ddlc-win.zip* file and double-click the *DDLC-1.1.1-pc* folder. Copy its contents to the folder you made in the Ren'Py SDK folder.
 - a. MacOS Users with a *ddlc-mac* folder
 - i. Copy the contents inside the *ddlc-mac* folder to the folder you made in the Ren'Py SDK folder.
 - b. MacOS Users with a *ddlc-mac* ZIP file
 - i. Open the *ddlc-mac.zip* file and copy its contents to the folder you made in the Ren'Py SDK folder.
2. Open the *mod template ZIP file* and copy its contents to the folder you made in the Ren'Py SDK folder. Accept any replaces.
 - a. MacOS Users with a *DDLCModTemplate-2.4.X* folder
 - i. Copy the contents inside the *DDLCModTemplate-2.4.X* folder to *DDLC.app/Contents/Resources/autorun*. Accept any replaces.
 - b. MacOS Users with a *DDLCModTemplate-2.4.X* ZIP file
 - i. Open the *DDLCModTemplate-2.4.X.zip* file and copy its contents to *DDLC.app/Contents/Resources/autorun*. Accept any replaces.
3. Open the *rpatool-master* ZIP file and double click the *rpatool-master* folder. Copy *rpatool* to the *game* folder.
 - a. MacOS Users with a *rpatool-master* folder
 - i. Copy *rpatool* from within the *rpatool-master/rpatool-master* folder to *DDLC.app/Contents/Resources/autorun/game*.
 - b. MacOS Users with a *rpatool-master* ZIP file
 - i. Open the *rpatool-master* ZIP file and double click the *rpatool-master* folder inside the ZIP. Copy *rpatool* to *DDLC.app/Contents/Resources/autorun/game*.
4. Run DDLC via DDLC.exe, DDLC.app, or via the Linux terminal by typing `./DDLC.sh` in the directory it is in.
5. If the game runs without any errors, you can exit the game and proceed to make your mod and follow the rest of the guide to setting up your mod for Android.
 - a. If DDLC doesn't start for the first time, try launching the game again. If that still doesn't work, look for any traceback or error.txt files, else report the issue to the GanstaKingofSA repository.

Upgrading Ren'Py/The Template

Upgrading from the Monika-After-Story/RationalPi Template (Versions 0.1.0 to 1.1.2)

This update has major code changes to be like DDLC which the Monika-After-Story/RationalPi template lacks; especially the android code needed to run the game on Android. It is highly suggested to move all your existing code to the 2.4.X files to retain all DDLC functionality, your past work, and have Android compatibility in your mod.

Upgrading from Versions 2.1.X to 2.3.1 (includes 2.3.1-uX versions)

Due to the Android code, some functionality will require a transfer of code to 2.4.X. I suggest transferring the code you have done to the 2.4.X files to retain your past work and have Android compatibility in your mod.

Upgrading from Versions 2.4.0 to 2.4.5 (for templates above 2.4.5)

Templates above 2.4.5 shouldn't be affected too much as the Android code in the template has just been optimized and fixed slightly. If you do plan to upgrade, please note the following.

- *package_name* in options.rpy has become obsolete in favor of environment paths.
- script-poemgame, splash, script, options, and definitions.rpy have changed a bit with its Android code; similar to options.rpy.

Android by following a DDLC Guide

Some DDLC guides will explain that in order to make your first DDLC mod, you will have to download some decompiled scripts or show you how to decompile the whole game. This, however, will not make your mod work on Android as it needs Android specific code for the game to run properly. Rely on the mod template files itself and this guide to help you make your mod work on Android.

Android under Ren'Py 7.3.5 - 7.4.4 or 7.4.6-7.4.8

Due to BinTray shutting down, Ren'Py versions older than 7.4.5 will not be able to compile mods on Android. There is no solution to resolve this problem apart from upgrading to 7.4.5 for now.

Using Ren'Py 7.4.6 - 7.4.8 will not work as they break DDLC heavily with it's transforms. Templates after 2.4.5 may lock the template from modding at all. Ren'Py 7.4.9 and higher will work only under Version 2.4.9 of the mod template.

Building the Mod for Android

Read the steps carefully before doing anything. This section assumes you followed the steps in **Installation Steps** and ready to publish your mod.

1. Open Terminal, Command Prompt or PowerShell in the game folder and run the following commands. Make sure you backup your mod files before you run them or they might get overridden with DDLC's own files.

```
python rpatool -x audio.rpa
```

```
python rpatool -x fonts.rpa
```

```
python rpatool -x images.rpa
```

- a. If you get a *Python was not found* error, see **Notes** Section F for more information.
 - b. If you get a *'python' is not recognized* error, then Python is not installed properly. Refer to **Setting Up the Template** for more information.
2. Start Ren'Py Launcher and select your mod from the Projects sidebar, then click *Android*.
 3. You will be asked to download RAPT. Select *Yes* and follow the prompts on-screen.
 4. Click *Android* again, then click *Install SDK and Select Keys*. It will start to download the necessary files it needs.
 - a. If the button is grayed out, then you are missing Temurin 8. Refer to **Setting Up the Template** for more information.
 5. Click *Configure* and follow the steps below to setup the mod configuration for Android.
 - a. For Full Name, set it as your mod name.
 - b. For Short Name, set it to your mod name's abbreviation.
 - c. For Package Name, use *com.* then your name with a period and the build name of your mod like such

```
com.[yourname].[buildname]
```

- d. For Application Version, set it to the version number of your mod.
 - e. For Version Code, leave it as is.
 - f. (For 7.4.5) For RAM Allocation to Gradle, leave it as-is.
 - g. For Display, set it to *landscape orientation*.
 - h. For app store in-app purchasing, select *Neither*.
 - i. Select *No* to creating an expansion APK.
 - j. (For 6.99.12.4) For version of Android, select *Android 4.0*.
 - k. Select *No* to allowing the mod to access the internet.
 - l. (For 7.4.5) Select *Yes* to automatically update your project.
6. Select *Build Package* and your mod should compile to a APK to be used on Android.
 - a. For Ren'Py 6, your mod will be compiled as the following file.

| `[ModName]-[Version]-release.apk`

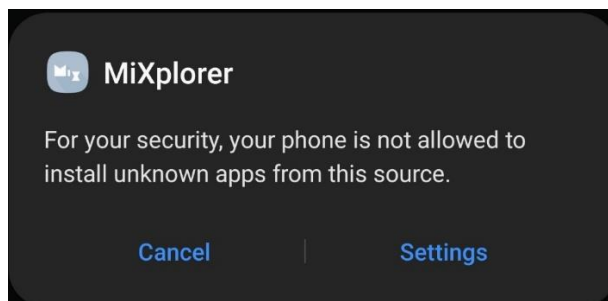
- b. For Ren'Py 7, your mod will be compiled into 4 APK files. Use the APK with the following name.

| `com.[yourname].[buildname]-[version]-universal-release.apk`

- c. If you encountered a *java.lang.OutOfMemoryError* when building your mod, see **Notes** Section E for more information.
- 7. Copy the APK over to your phone and make sure it runs as intended.
 - 8. If your mod runs as intended, upload it to Google Drive, Mega, etc., and release it to the public!
 - a. Make sure to add to your post 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*. Refer to **Notes** Sections A to C for more information.

Notes

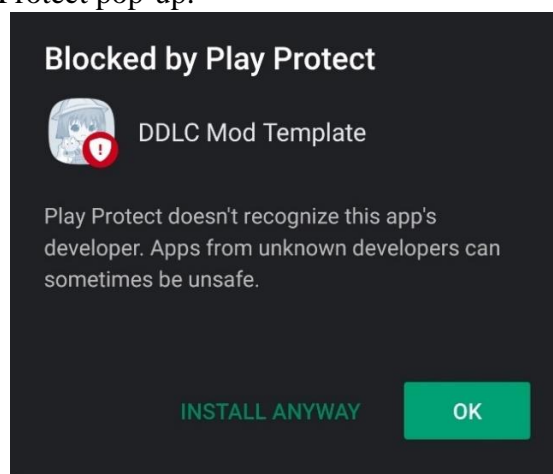
- a. When installing your mod either via a file manager or by downloading off your browser, etc. you may get this pop-up by Android or something similar.



To fix this turn on Unknown Sources on your device. Android will warn you that installing apps from unknown sources may be malicious. Select either *OK* or (for some devices) *Install Just Once* then install the app to your device.

This step will differ from device to device. Consult Google for more information on turning on unknown sources for your device.

- b. When installing your mod either via a file manager or by downloading off Chrome, etc. you may get this Play Protect pop-up.



Tap Install Anyway and the installation should proceed as normal.

- a. Alternatively, you may get a pop-up saying the app is built for a old version of Android. Tap Install Anyways to continue the install as normal.
- c. While running the game you might get a pop-up that asks whether to send your over to Google for analysis. You can choose either option presented as this is just a Google Security Message.

- d. If Temurin 8 and 11+ are installed on the same computer, it may cause an error in Ren'Py in which it cannot build your mod due to outdated code.

Make sure that only Temurin 8 is installed (re-install it if Temurin 11+ was uninstalled) or build your mod on another device or on a virtual machine with just Temurin 8 installed on it.

- e. Sometimes when building your mod, you may run into an error saying *Caused by: java.lang.OutOfMemoryError*. This error is caused due to Java using all the space allocated to it while building your mod which is 3 GB.

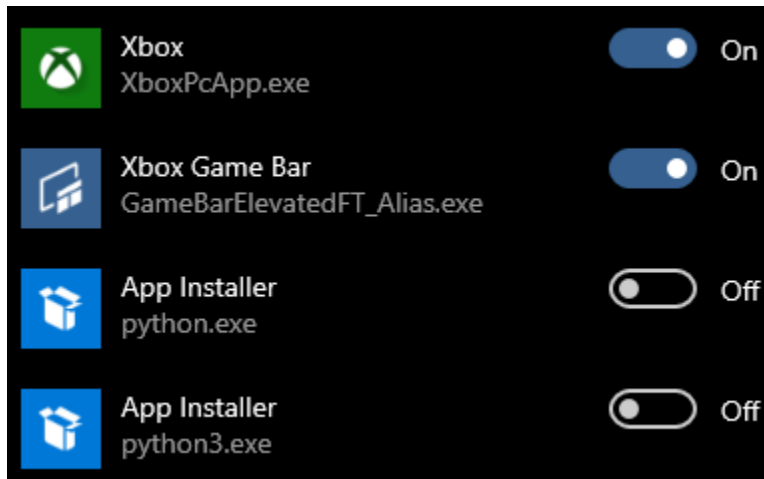
To fix this, go into *rapt/project* and open *gradle.properties* with Visual Studio Code. Increase the three in *-Xmx3g* by one and try again. Do not allocate more memory than what your system comes with.

- If you are running Ren'Py 7.4.5 or DDMM/DDMMaker 1.2.6 or higher, you can fix this within the Android Menu via *Configure* (Ren'Py Launcher and later DDMM/DDMMaker versions) or *Change RAPT Settings* (DDMM/DDMMaker 1.2.6).

- f. When executing the RPATool commands in Windows, you might get this error.

Python was not found; run without arguments to install from the Microsoft Store, or disable this shortcut from Settings > Manage App Execution Aliases.

To fix this, go to Windows Search and type *Manage App Execution Aliases*. In it, turn off App Installer for *python.exe*. Afterwards, re-open Command Prompt or PowerShell and type the commands in again.



FAQ

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

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

Q. A bunch of my scripts is broken.

Ren'Py on Android runs differently than on PC which can cause PC designed code to fail on Android. Refer to the template comments or the Ren'Py Documentation to migrate some code over to Android from PC.

Q. Does the package name have to be *com.[yourname].[modbuildname]*?

No. You can change it to be whatever (except the *modbuildname* part), but it must follow the Android naming scheme of *x.y.z* and be all lowercased.

Q. My GUI elements are all broken.

The GUI positions loaded is of DDLC but modified to work on mobile devices. The code for it is located at the bottom section of *gui.rpy*. Change the variables there to adjust your GUI to how it is in-game on PC.

Q. Isn't distributing the APK as such against Team Salvato's IPG?

No. Team Salvato acknowledges that the game assets are needed for mods to run on Android and allows it for the time being. However, they will keep an eye on how these APK files are distributed and this decision may be revoked at any time.

Q. What is the difference between this and JoiPlay?

JoiPlay emulates mods as if you were on a computer while this runs mods as an Android designed app. This means it is prone to less bugs and crashes compared to JoiPlay.

Q. How would I add my files that need to be read, written to or modified in *Android/data/com.[yourname].[modbuildname]* folder?

The mod template files show some examples on how these files work in the game under Android. Below are a few examples of code that writes files into the Android directory.

This code below shows an example of writing *CAN YOU HEAR ME.txt* to the game's Android directory.

```
python:
if renpy.android:
    # For Android, the try and excepts must be formatted like so with this but replace
    # hxppy thxughts.png with the file you want to write.
    ## try: renpy.file(os.environ['ANDROID_PUBLIC'] + "/hxppy thxughts.png")
    ## except: open(os.environ['ANDROID_PUBLIC'] + "/hxppy thxughts.png", "wb").write(renpy.file("hxppy thxughts.png").read())
    try: renpy.file(os.environ['ANDROID_PUBLIC'] + "/hxppy thxughts.png")
    except: open(os.environ['ANDROID_PUBLIC'] + "/hxppy thxughts.png", "wb").write(renpy.file("hxppy thxughts.png").read())
else:
    try: renpy.file(config.basedir + "/hxppy thxughts.png")
    except: open(config.basedir + "/hxppy thxughts.png", "wb").write(renpy.file("hxppy thxughts.png").read())
```

This code below shows an example of writing the character files into the *characters* folder if a Doki's character file is missing from within the Android directory.

```
import os
if renpy.android: #checks if the platform is android
    try:
        # writes character files if missing and correct playthrough to Android/data/[your mod]/characters
        if not os.access(os.environ['ANDROID_PUBLIC'] + "/characters/", os.F_OK):
            os.mkdir(os.environ['ANDROID_PUBLIC'] + "/characters")
        if persistent.playthrough <= 2:
            try: renpy.file(os.environ['ANDROID_PUBLIC'] + "/characters/monika.chr")
            except: open(os.environ['ANDROID_PUBLIC'] + "/characters/monika.chr", "wb").write(renpy.file("monika.chr").read())
        if persistent.playthrough <= 1 or persistent.playthrough == 4:
            try: renpy.file(os.environ['ANDROID_PUBLIC'] + "/characters/natsuki.chr")
            except: open(os.environ['ANDROID_PUBLIC'] + "/characters/natsuki.chr", "wb").write(renpy.file("natsuki.chr").read())
            try: renpy.file(os.environ['ANDROID_PUBLIC'] + "/characters/yuri.chr")
            except: open(os.environ['ANDROID_PUBLIC'] + "/characters/yuri.chr", "wb").write(renpy.file("yuri.chr").read())
        if persistent.playthrough == 0 or persistent.playthrough == 4:
            try: renpy.file(os.environ['ANDROID_PUBLIC'] + "/characters/sayori.chr")
            except: open(os.environ['ANDROID_PUBLIC'] + "/characters/sayori.chr", "wb").write(renpy.file("sayori.chr").read())
    except:
        pass
else:
    try:
        if not os.access(config.basedir + "/characters/", os.F_OK):
            os.mkdir(config.basedir + "/characters")
        if persistent.playthrough <= 2:
            try: renpy.file("../characters/monika.chr")
            except: open(config.basedir + "/characters/monika.chr", "wb").write(renpy.file("monika.chr").read())
        if persistent.playthrough <= 1 or persistent.playthrough == 4:
            try: renpy.file("../characters/natsuki.chr")
            except: open(config.basedir + "/characters/natsuki.chr", "wb").write(renpy.file("natsuki.chr").read())
            try: renpy.file("../characters/yuri.chr")
            except: open(config.basedir + "/characters/yuri.chr", "wb").write(renpy.file("yuri.chr").read())
        if persistent.playthrough == 0 or persistent.playthrough == 4:
            try: renpy.file("../characters/sayori.chr")
            except: open(config.basedir + "/characters/sayori.chr", "wb").write(renpy.file("sayori.chr").read())
    except:
        pass
```

This FAQ is not an exhaustive list of questions for DDLA Android mod development. For further questions, ask in the *#mod_help* channel or see the *#mod_faq* channel on the DDMC Discord linked [here](https://discord.gg/Gx7m9xE) or contact me in DM.

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

Acknowledgements

I want to acknowledge the following people for assistance, either voluntary or in-voluntary.

Tom Rothamel (PyTom) for Ren'Py, RAPT and Ren'Py Android functionality.

Dan Salvato for making DDLC and allowing us to mod his game. Also inspiration for the Android Guidelines of this guide.

My past projects to optimize the mod template with better code.

You for reading this whole guide.

Weiss Schnee for support. (Weiss: 😊)