



FREE eBook

LEARNING android-studio

Free unaffiliated eBook created from
Stack Overflow contributors.

#android-
studio

Table of Contents

About.....	1
Chapter 1: Getting started with android-studio.....	2
Remarks.....	2
Versions.....	2
Examples.....	3
Installation or Setup.....	3
Windows.....	3
Mac.....	4
Linux.....	4
Notes.....	5
Problem with downloading.....	5
Useful shortcuts.....	5
Navigation Shortcuts.....	5
Navigation Shortcuts - Mac OS X.....	6
Programming Shortcuts.....	6
Preview Different Screen Size (Devices) and Orientations.....	6
1. Preview Different Devices.....	6
2. Switching Orientation.....	8
Use your favorite tool shortcuts in Android Studio.....	10
Chapter 2: Android Studio optimization.....	11
Parameters.....	11
Examples.....	11
Customize the VM Option.....	11
Windows:.....	11
Mac:.....	11
Linux:.....	11
Default setting for Android Studio 64-bit.....	11
Optimized Setting.....	12
Chapter 3: Android Studio Tips and Tricks.....	13

Examples.....	13
Emulator for testing.....	13
Custom Live Template.....	13
Example :.....	13
To achieve this purpose , you should add a custom Live Template:.....	14
Use Custom Code Styles, Share with other Team Members and Auto Format with Shortcut.....	17
Chapter 4: Android Studio updates.....	20
Examples.....	20
Updating Android Studio in Ubuntu.....	20
Android Studio update channels.....	20
Overview.....	20
Selecting an Update Channel.....	20
Credits.....	22

About

You can share this PDF with anyone you feel could benefit from it, downloaded the latest version from: [android-studio](#)

It is an unofficial and free android-studio ebook created for educational purposes. All the content is extracted from [Stack Overflow Documentation](#), which is written by many hardworking individuals at Stack Overflow. It is neither affiliated with Stack Overflow nor official android-studio.

The content is released under Creative Commons BY-SA, and the list of contributors to each chapter are provided in the credits section at the end of this book. Images may be copyright of their respective owners unless otherwise specified. All trademarks and registered trademarks are the property of their respective company owners.

Use the content presented in this book at your own risk; it is not guaranteed to be correct nor accurate, please send your feedback and corrections to info@zzzprojects.com

Chapter 1: Getting started with android-studio

Remarks

Android Studio is the official IDE by Google for Android app development, based on IntelliJ IDEA. [\[1\]](#)

It utilizes a Gradle-based build system through the Android Plugin for Gradle. [\[2\]](#)

Versions

Version	Release Date
0.1.x	2013-05-01
0.2.x	2013-07-01
0.3.2	2013-10-01
0.4.2	2014-01-01
0.4.6	2014-03-01
0.5.2	2014-05-01
0.8.0	2014-06-01
0.8.6	2014-08-01
0.8.14	2014-10-01
1.0	2014-12-01
1.0.1	2014-12-02
1.1.0	2015-02-01
1.2.0	2015-04-01
1.2.1	2015-05-01
1.2.2	2015-06-01
1.3.0	2015-07-01
1.3.1	2015-08-01

Version	Release Date
1.3.2	2015-08-02
1.4.0	2015-09-01
1.4.1	2015-10-01
1.5.0	2015-11-01
1.5.1	2015-12-01
2.0	2016-04-01
2.1.0	2016-04-02
2.1.1	2016-05-01
2.1.2	2016-06-01
2.1.3	2016-08-01
2.2.0	2016-09-01
2.3.0	2017-03-02
2.3.1	2017-04-02

Examples

Installation or Setup

Windows

1. Verify that you have the correct JDK. You can check it by opening command prompt (press `windows key` and write `cmd`). In the command prompt type `javac -version`, this will show the current version of JDK installed on your matching or an error* if Java is missing. If the JDK is not available or the version is lower than 1.8, download the [Java SE Development Kit 8](#).
2. Download the latest [Android Studio](#).
3. Launch the downloaded `.exe` file.
4. Follow the wizard to install Android Studio
5. After installation completes, open Android Studio from the shortcut that has been made on Desktop during the installation process.

6. When you open Android Studio for first time, it may ask you to copy your previous settings, but as it is your first time you can simply choose not to copy anything.
7. Then Android Studio ask to download the required API's to develop Android apps. Download those, after downloading the IDE will open and you will be able to write your first app.

Mac

1. Verify that you have the correct JDK. You can check it by opening terminal (press `command + space` and write `terminal`). In the command line type `javac -version`, this will show the current version of JDK installed on your matching or an error* if Java is missing. If the JDK is not available or the version is lower than 1.8, download the [Java SE Development Kit 8](#). **
2. Download the latest [Android Studio](#).
3. Launch the downloaded `.dmg` file.
4. Drag and drop Android Studio into the Applications folder, then launch Android Studio.
5. Open Android Studio.
6. When you open Android Studio for first time, it may ask you to copy your previous settings, but as it is your first time you can simply choose not to copy anything.
7. Then Android Studio ask to download the required API's to develop Android apps. Download those, after downloading the IDE will open and you will be able to write your first app.

Linux

1. Verify that you have the correct JDK. You can check it by opening terminal (press `command + space` and write `terminal`). In the command line type `javac -version`, this will show the current version of JDK installed on your matching or an error* if Java is missing. If the JDK is not available or the version is lower than 1.8, download the [Java SE Development Kit 8](#). **
2. Download the latest [Android Studio](#).
3. Unzip/extract Android Studio in a specific folder.
4. Open terminal and go the path where you have extracted the Android Studio. (Then, use `cd` command to go inside the Android Studio folder.) After going in we need to go inside `bin` folder so again,use command `cd bin` and enter.
5. Now we need to change the mod of our required file i.e `studio.sh` to do so enter command `sudo chmod 777 -R studio.sh` , press enter and write your password(if any) and enter. (Also you can see the list of files present inside bin by command `ls`).
6. After changing mod we just have to run the `.studio.sh` file to do so enter command

```
./studio.sh
```

7. When you run above command Android Studio launch for first time, it may ask you to copy your previous settings, but as it is your first time you can simply choose not to copy anything.
8. Then Android Studio ask to download the required API's to develop Android apps. Download those, after downloading the IDE will open and you will be able to write your first app.

You may encounter "unable to run mksdcard SDK tool" when you are installing android studio in 64 bit ubuntu os because studio requires some 32 binaries. To overcome this error finish and close Android Studio & go to terminal and run `sudo apt-get install lib32z1 lib32ncurses5 lib32bz2-1.0 lib32stdc++6`. Once installation is completed for these binaries again go back to step 6 and resume installation process.

Notes

- If you have already installed JDK and still getting then make sure you have set `JAVA_HOME` in your `System Variables`. You can check this [answer](#) on how to setup one.
- There are, however, known stability issues in Android Studio on Mac when using JDK 1.8. Until these issues are resolved, you can improve stability by downgrading your JDK to an older version (but no lower than JDK 1.6).
- While the Android Studio download completes, verify which version of the **JDK** you have: open a command line and type `javac -version`. If the JDK is *not* available or the version is lower than 1.8, download the [Java SE Development Kit 8](#).
- If you are running a 64-bit version of **Ubuntu**, you need to install some 32-bit libraries with the following commands:

```
sudo dpkg --add-architecture i386
sudo apt-get update
sudo apt-get install lib32z1 lib32ncurses5 libbz2-1.0:i386 lib32stdc++6
```

- If you are running 64-bit **Fedora**, the command is:

```
sudo yum install zlib.i686 ncurses-libs.i686 bzip2-libs.i686
```

Problem with downloading

- If you would face any issue like message that your system is not compatible with Android Studio (it may happen when you're using web browser different than Chrome) download Android Studio from: <http://tools.android.com/>

That's it!

Useful shortcuts

Navigation Shortcuts

Go to class	Ctrl+N
Go to file	Ctrl + Shift + N
Navigate open tabs	ALT + Left-Arrow; ALT + Right-Arrow
Lookup recent files	CTRL + E
Go to line	CTRL + G
Navigate to last edit location	CTRL + SHIFT + BACKSPACE
Go to declaration	CTRL + B
Go to implementation	CTRL + ALT + B
Go to source	F4
Go to super Class	CTRL + U
Show Call hierarchy	Ctrl + Alt + H
Search in path/project	CTRL + SHIFT + F

Navigation Shortcuts - Mac OS X

Go to line number	CMD + L
-------------------	---------

Programming Shortcuts

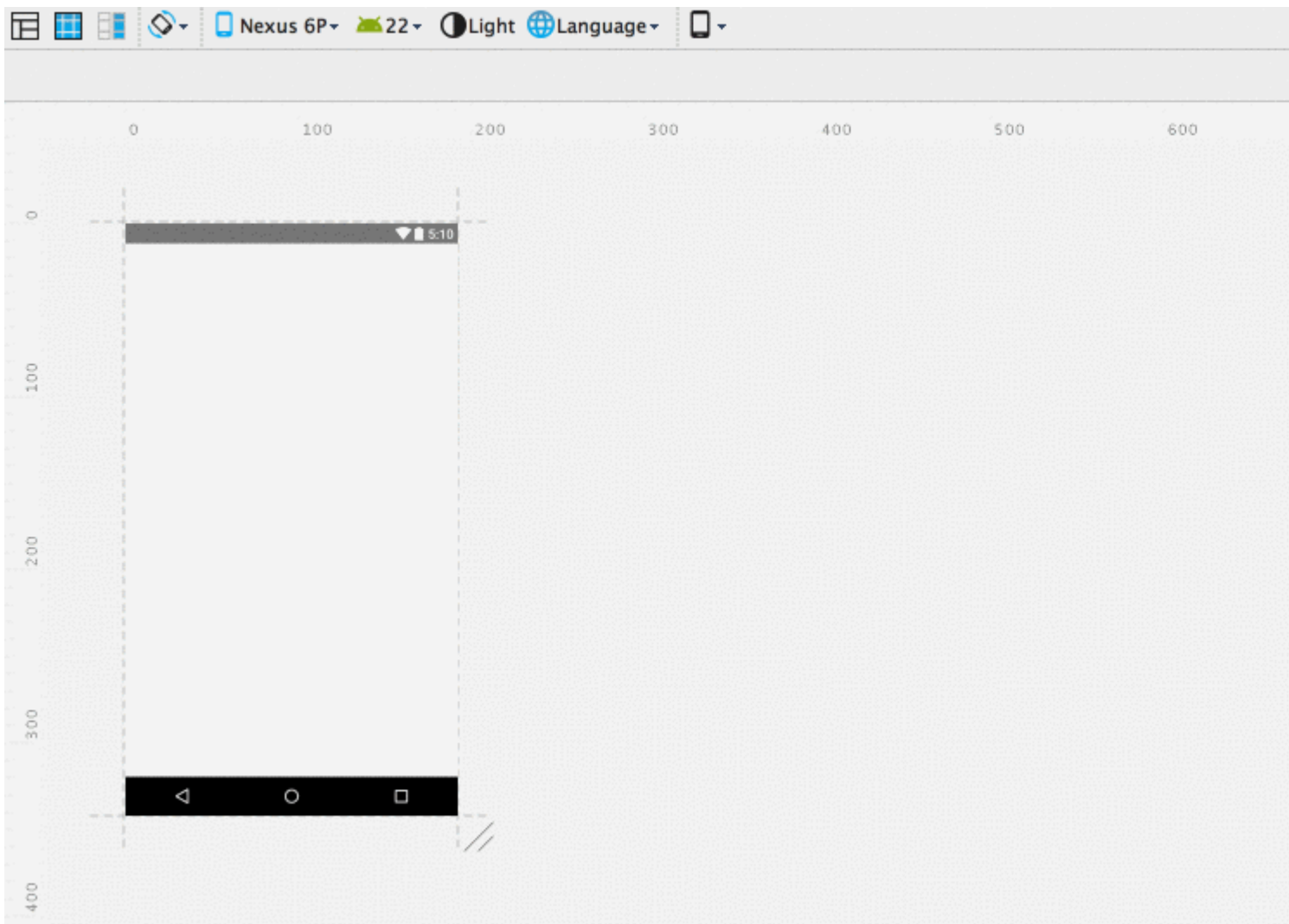
Reformat code	CTRL + ALT + L
Optimize imports	CTRL + ALT + O
Code Completion	CTRL + SPACE
Issue quick fix	ALT + ENTER
Surround code block	CTRL + ALT + T
Rename and refactor	Shift + F6
Line Comment or Uncomment	CTRL + /
Block Comment or Uncomment	CTRL + SHIFT + /
Go to previous/next method	ALT + UP/DOWN
Show parameters for method	CTRL + P
Quick documentation lookup	CTRL + Q
Project	Alt+1
Version Control	Alt+9
Run	Shift+F10
Debug	Shift+F9
Android Monitor	Alt+6
Return to Editor	Esc
Hide All Tool Windows	CTRL +Shift+F12
Auto generate code (constructor, getter/setter etc)	Alt+insert
Code completion	CTRL+Space
Smart code completion (by expected type)	CTRL+Shift+Space
Show quick fix	CTRL+Enter
Duplicate Line	Ctrl+D
Delete Line	Ctrl+Y

Preview Different Screen Size (Devices) and Orientations

2.1.3

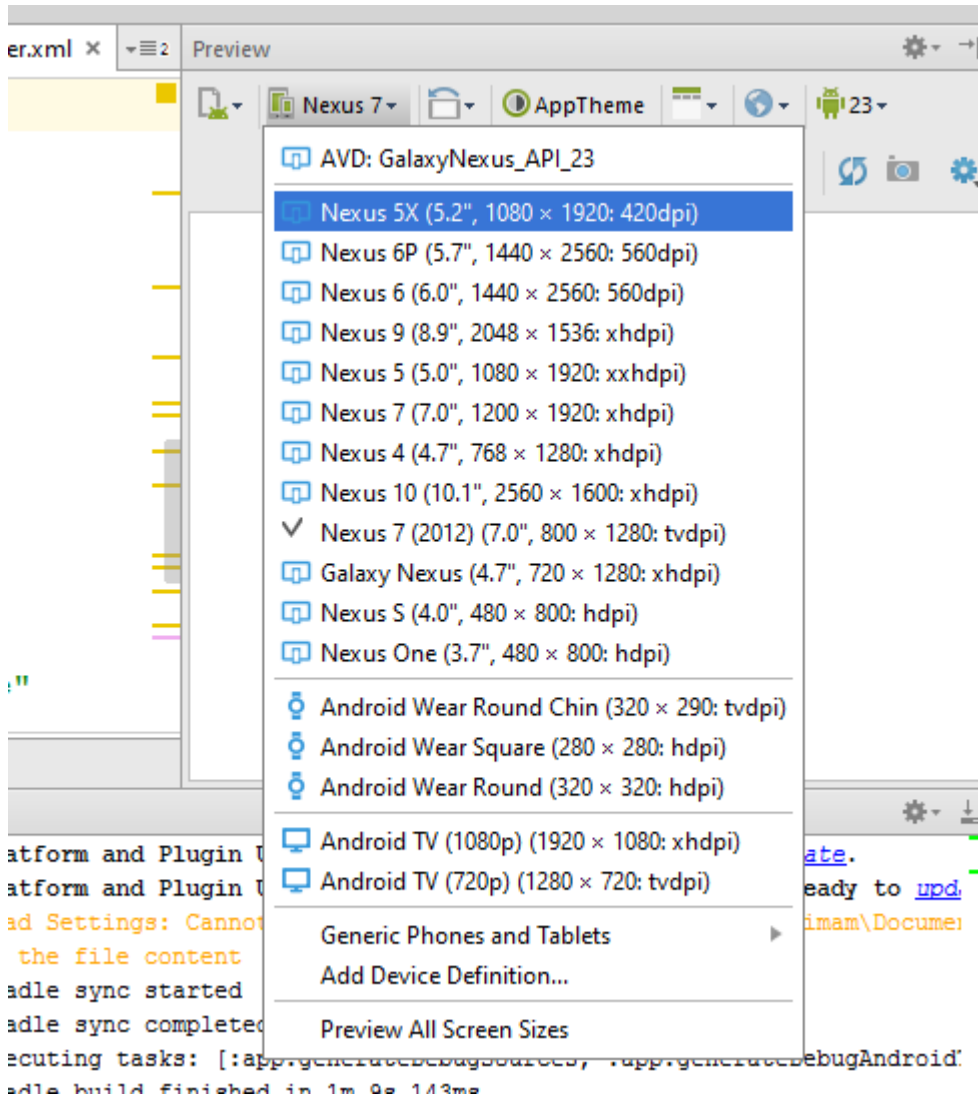
1. Preview Different Devices

There is a preview panel at the right of the android studio. In this panel there is a button with device name with which you are previewing the UI of your app like this .



Click on small dropdown indicator of this and a floating panel will appear with all the predefined devices. You can select any of them to preview your App UI with different devices mentioning their screen sizes.

Check this image



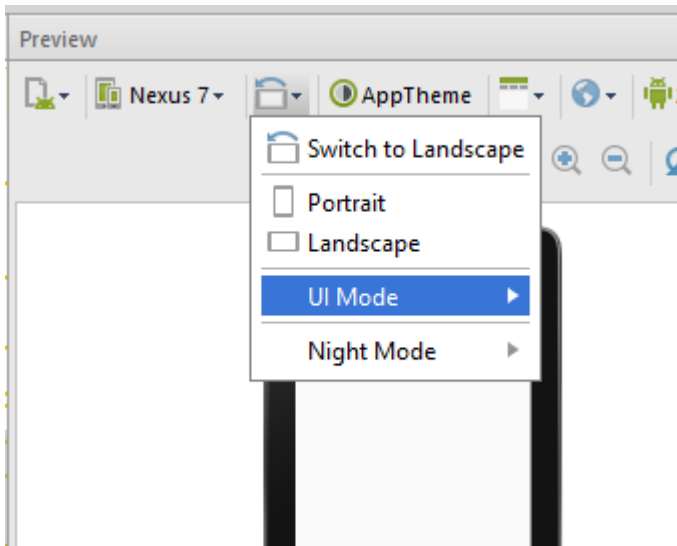
2. Switching Orientation

And Next to this button there is another button like



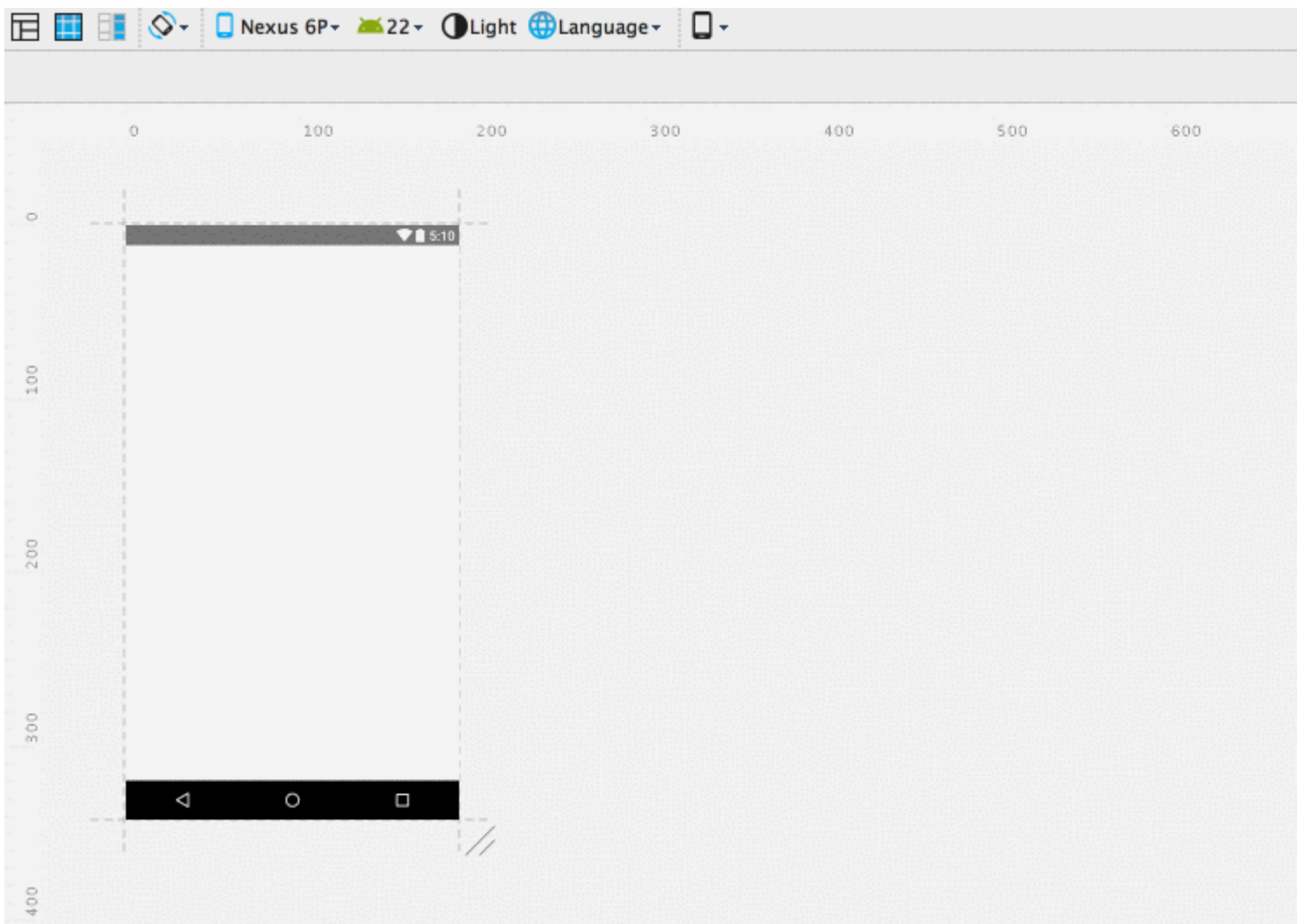
Clicking to its dropdown a floating panel will appear with some options like `portrait` , `landscape` etc. select one of them to preview in different orientations.

Check the this image



2.2.0

The option to preview all screen sizes is not available from Android Studio 2.2 Beta 2. However, you can resize the preview screen to see how your layout would look on different screen sizes, as shown in the attached.



Check: [Android Studio 2.2 Preview all screen sizes missing](#)

Use your favorite tool shortcuts in Android Studio

Go to File > Settings > Keymap and select the Keymaps option from:

- Mac OS X
- Emacs
- Visual Studio
- Eclipse
- Netbeans
- Jbuilder

and others, to map the shortcuts to the wanted tool ones.

Read [Getting started with android-studio online](https://riptutorial.com/android-studio/topic/830/getting-started-with-android-studio): <https://riptutorial.com/android-studio/topic/830/getting-started-with-android-studio>

Chapter 2: Android Studio optimization

Parameters

Parameter	Detail
Xms	Initial memory allocate
Xmx	Max memory allocate

Examples

Customize the VM Option

You can override the default `vmoptions` with your own personal settings by choosing **Help > Edit Custom VM Options** from the Android Studio toolbar. This will create a local copy which you are free to edit.

Alternatively, you can edit the default `vmoptions` directly using the paths given below. Note that this method is not recommended, and your changes may be overwritten when updating Android Studio.

Windows:

```
%USERPROFILE%\.{FOLDER_NAME}\studio.exe.vmoptions and/or  
%USERPROFILE%\.{FOLDER_NAME}\studio64.exe.vmoptions  
%USERPROFILE%\.{FOLDER_NAME}\idea.properties
```

Mac:

```
~/Library/Preferences/{FOLDER_NAME}/studio.vmoptions  
~/Library/Preferences/{FOLDER_NAME}/idea.properties
```

Linux:

```
~/.{FOLDER_NAME}/studio.vmoptions  
~/.{FOLDER_NAME}/studio64.vmoptions  
~/.{FOLDER_NAME}/idea.properties
```

Default setting for Android Studio 64-bit

```
-Xms128m
```

```
-Xmx750m
-XX:MaxPermSize=350m
-XX:ReservedCodeCacheSize=96m
-ea
-Dsun.io.useCanonCaches=false
-Djava.net.preferIPv4Stack=true
-Djna.nosys=true
-Djna.boot.library.path=

-Djna.debug_load=true
-Djna.debug_load.jna=true
-Djsse.enableSNIExtension=false
-XX:+UseCodeCacheFlushing
-XX:+UseConcMarkSweepGC
-XX:SoftRefLRUPolicyMSPerMB=50
-Didea.platform.prefix=AndroidStudio
-Didea.paths.selector=AndroidStudio
```

Optimized Setting

```
-Xms1024m
-Xmx4096m
-XX:MaxPermSize=1024m
-XX:ReservedCodeCacheSize=256m
-ea
-Dsun.io.useCanonCaches=false
-Djava.net.preferIPv4Stack=true
-Djna.nosys=true
-Djna.boot.library.path=

-Djna.debug_load=true
-Djna.debug_load.jna=true
-Djsse.enableSNIExtension=false
-XX:+UseCodeCacheFlushing
-XX:+UseConcMarkSweepGC
-XX:SoftRefLRUPolicyMSPerMB=50
-Didea.platform.prefix=AndroidStudio
-Didea.paths.selector=AndroidStudio
```

Read Android Studio optimization online: <https://riptutorial.com/android-studio/topic/3247/android-studio-optimization>

Chapter 3: Android Studio Tips and Tricks

Examples

Emulator for testing

If there are no specific needs, choose `x86_64` system images with Google APIs to create the emulator.

It works way faster than `armeabi-v7a` on intel x86 x64 based computers.

There are some SDK libraries compiled and designed with ARM architecture.

If you try to install them on Intel based emulators you'll get an error message saying

```
"INSTALL_FAILED_NO_MATCHING_ABIS"
```

You can go for a 3rd party emulator like [Genymotion](#) or [Visual Studio's standalone Android Emulator](#) on such occasions.

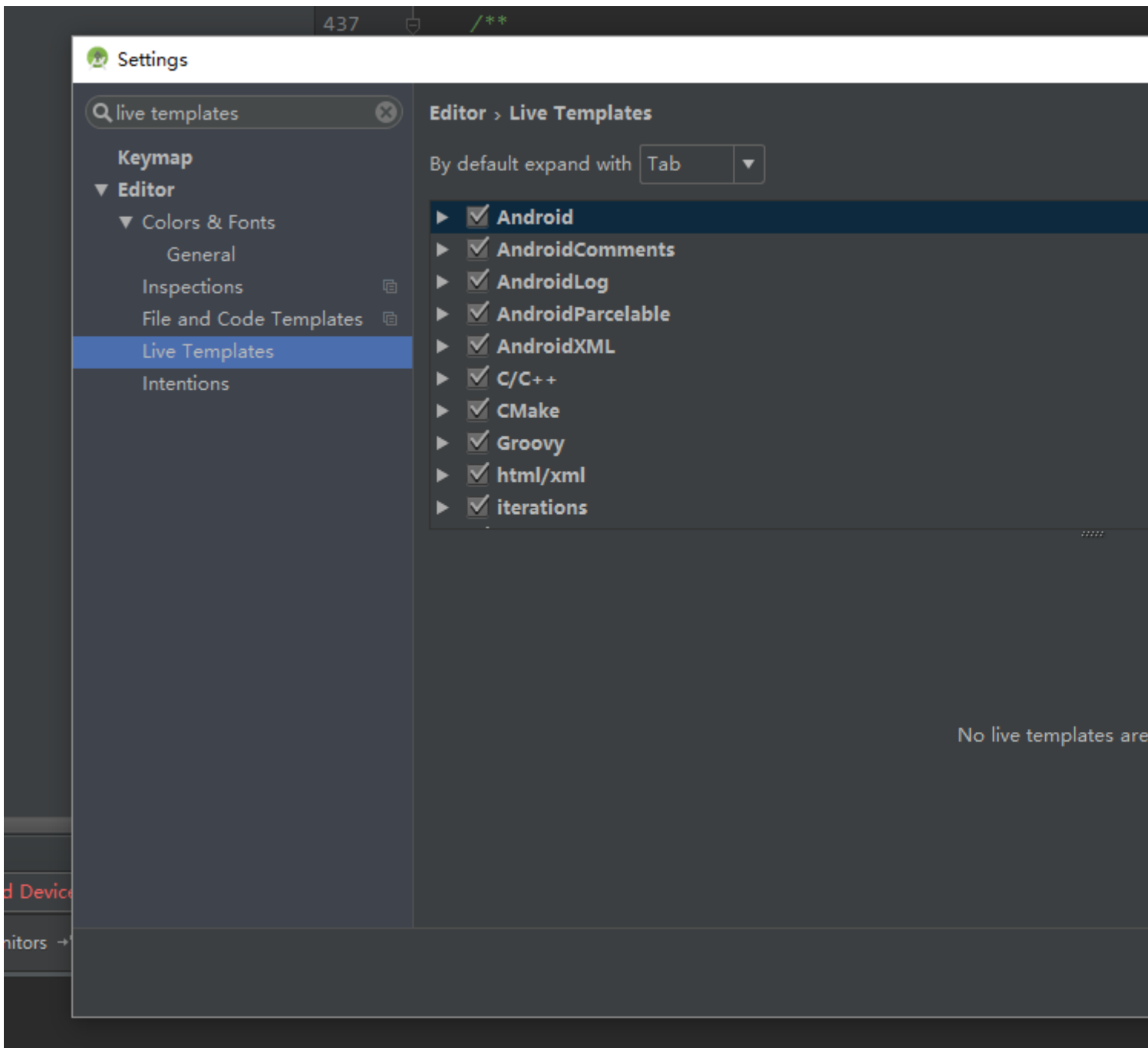
Custom Live Template

Example :


```
public class MainActivity extends AppCompatActivity {  
  
    @Override  
    protected void onCreate(Bundle savedInstanceState) {  
        super.onCreate(savedInstanceState);  
        setContentView(R.layout.activity_main);  
        Toolbar toolbar = (Toolbar) findViewById(R.id.toolbar);  
        setSupportActionBar(toolbar);  
        fastFinish();  
    }  
  
    public void fastFinish(){  
    }  
}
```

To achieve this purpose , you should add a custom Live Template:

1. Open Settings [Ctrl + Alt + s]
2. Type "Live Templates" in the Top-Left search bar.
3. Click the + to add a "Template Group" and type a group name(eg: MyTemplate) to continue

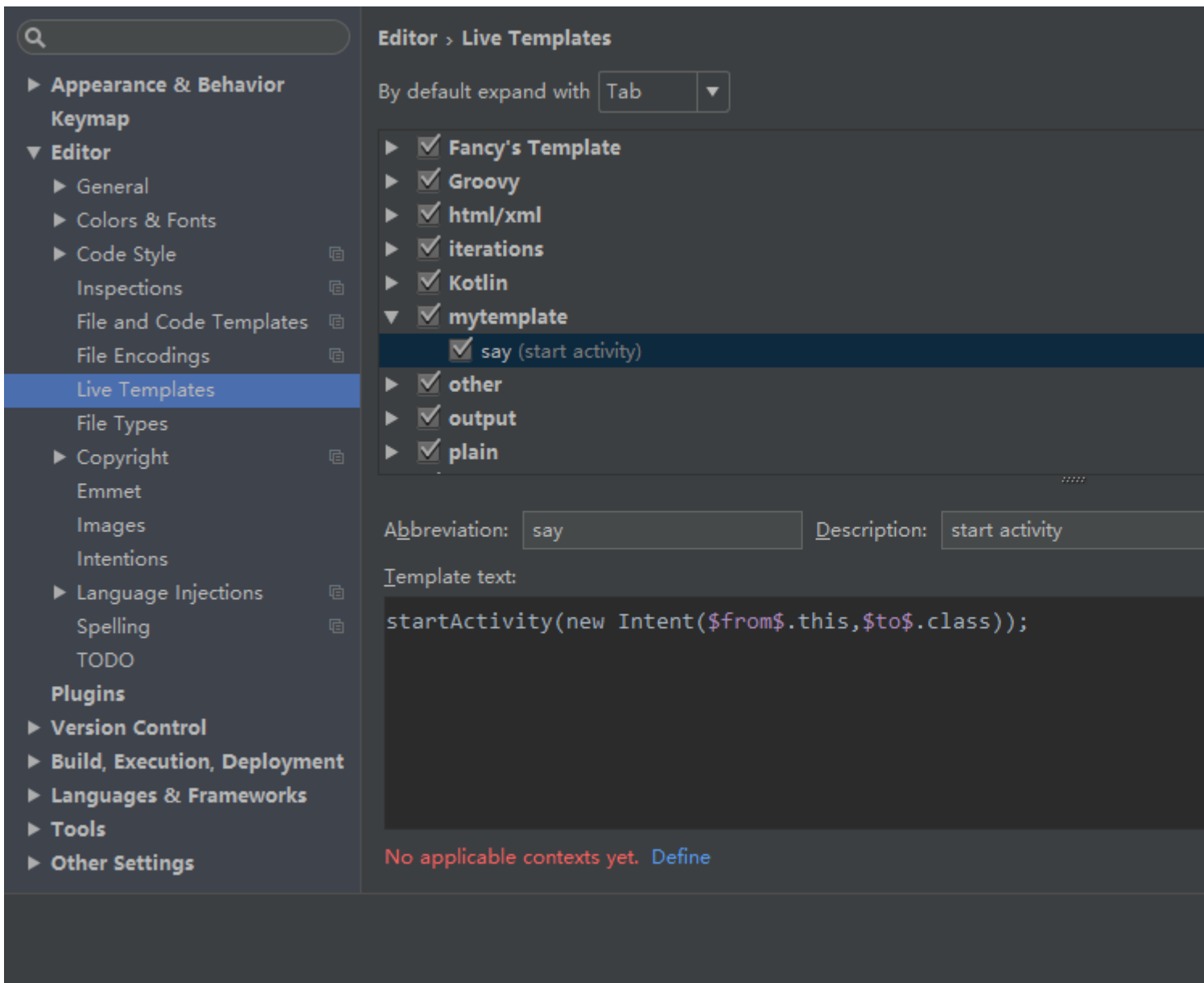


4. Choose your custom template group(eg: MyTemplate), then click + and choose "Live Template".

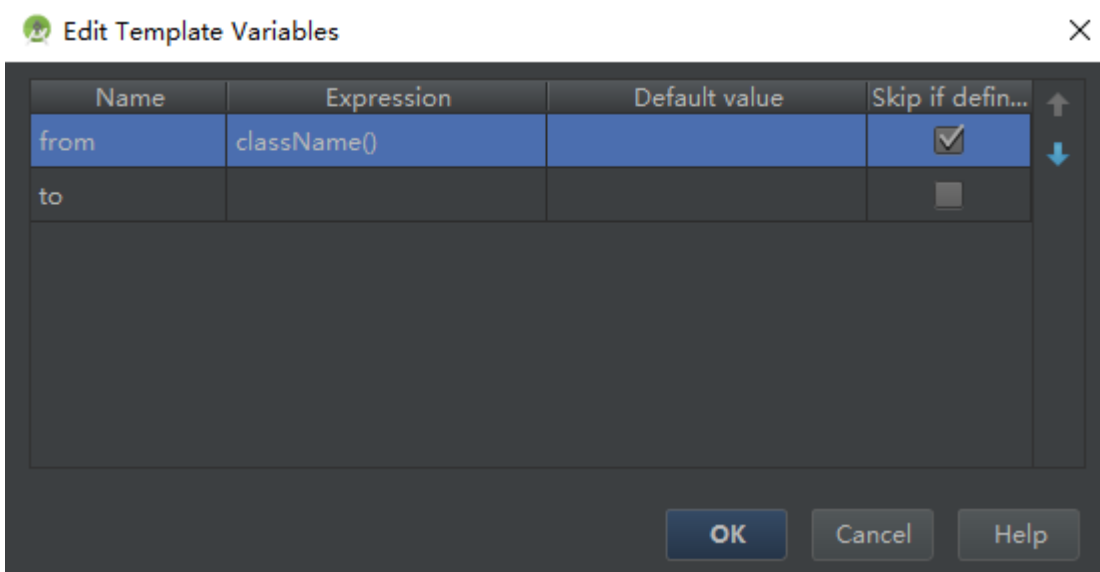
5. I will take "say" for example:

```
say -> startActivity(new Intent($from$.this,$to$.class));
```

6. Type the key that you want to simplify in "Abbreviation" (eg: say),and type the statement in the "Template Text" (eg :startActivity(new Intent(\$from\$.this,\$to\$.class));)



7. Click the "define" in the Left-Bottom and choose the situation(eg: Java)
8. Click the "Edit variables",define the expression.(see detail:[Edit Template Variables Dialog](#))



9. Click "OK" and "Apply". And try to type "say" in your editor.

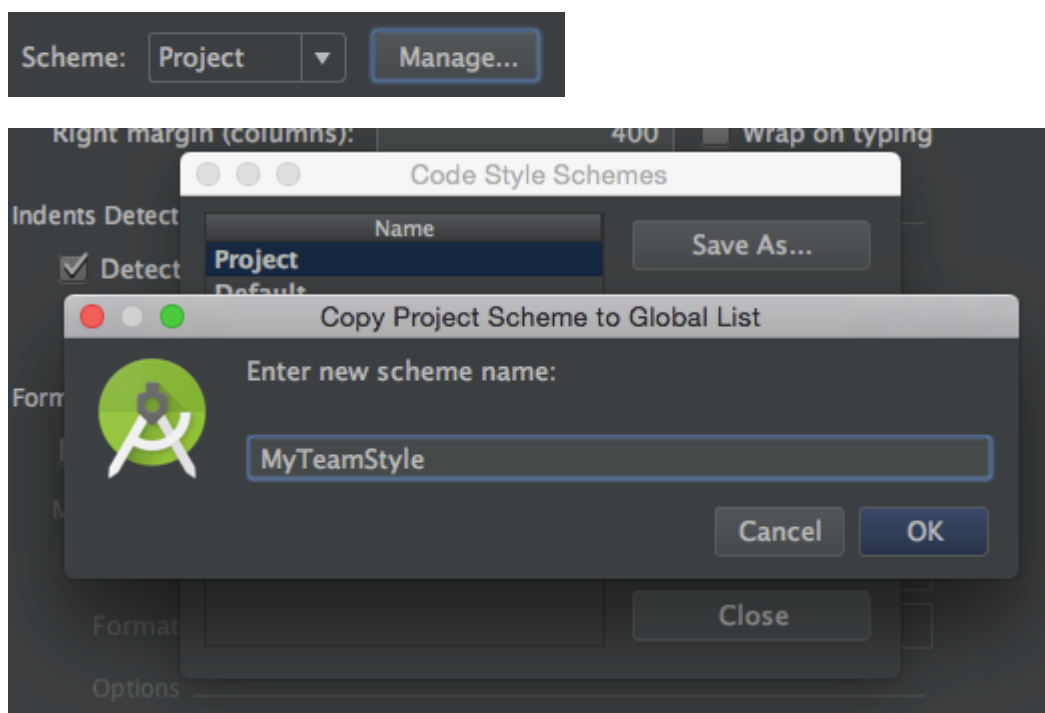
Use Custom Code Styles, Share with other Team Members and Auto Format with Shortcut

It's possible to have your own custom code styles, share them with other team members and use a shortcut to auto format the code in a file.

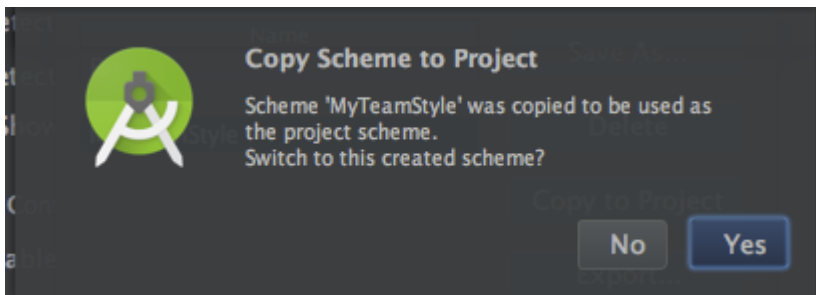
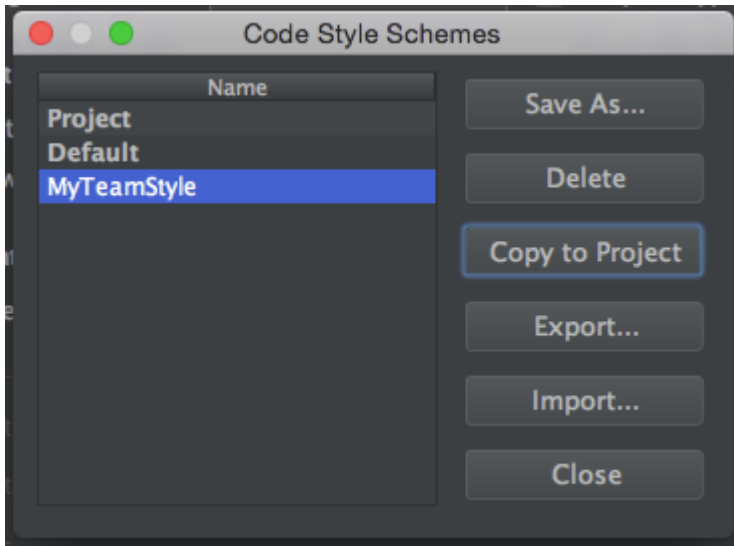
To create your own custom code style, go to: Preferences -> Editor -> Code Style

There are some general code style settings here. You can also select the language (Java for Android) and set the parameters as you see fit. There's a *lot* of settings.

Once you've done this, save your code style. This is just for safe keeping.



Once you've saved it, select "Copy to Project" from the "Manage" dialog. You will be asked whether you want to switch to this created scheme. Answer yes.



Once this is done, close the Code Style preferences dialog.

Now, verify that your code style settings have been saved in: `.idea/codeStyleSettings.xml`

Ensure that this file is not ignored in your version control system so that you can share it with your teammates.

Once your teammates have this file, they should also have the same settings.

Now, when editing a file, you can format the code by selecting:

Code -> Reformat Code

Shortcuts for Reformat Code (taken from [this](#) answer - see answer for details on resolving issues with shortcut in Ubuntu):

Win

`Ctrl + Alt + L`

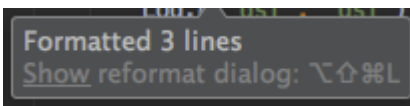
Linux:

`Ctrl + Shift + Alt + L`

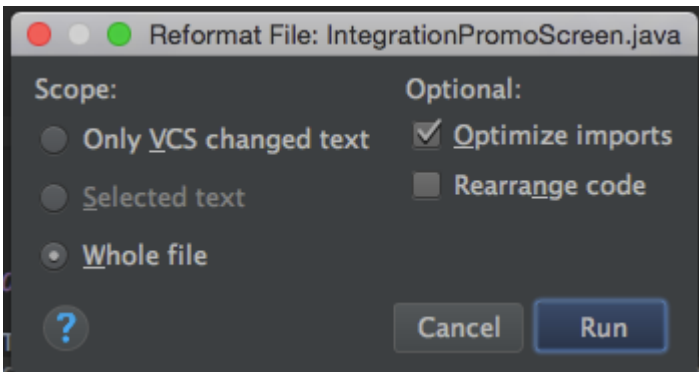
Mac:

`Option + Command + L`

When you perform a code reformat, a small dialog should popup informing you of the changes made. You can click "Show" in this dialog to bring up the "Reformat File Dialog".



You can also bring up this dialog from the Code menu and its corresponding shortcut.



Be warned that "Only VCS changed text" doesn't always work depending on how the code has been edited (it may ignore a rule if part of the code has not been edited).

You can also select text and show the reformat code dialog to format only the select text.

Read [Android Studio Tips and Tricks](https://riptutorial.com/android-studio/topic/2228/android-studio-tips-and-tricks) online: <https://riptutorial.com/android-studio/topic/2228/android-studio-tips-and-tricks>

Chapter 4: Android Studio updates

Examples

Updating Android Studio in Ubuntu

If you are able to download an update of Android Studio, but after it restarts nothing happens, check out the following example:

1. After the patch was downloaded and Android Studio closed, open the terminal
2. Go to your android studio folder, e.g. `cd ~/android-studio`
3. Go to `bin` subfolder: `cd bin`
4. Make sure your `studio.sh` file has run permissions: `chmod +x studio.sh`
5. Run Android Studio from here: `./studio.sh`

After that Android Studio will find the patch and install it. Then you may close Android Studio and run it the regular way (whatever way you prefer, I run it from Launcher pad).

Android Studio update channels

Overview

Android Studio's built-in update mechanism can be set to receive updates through any one of these **four** channels:

- **Canary:** Bleeding edge, released about weekly. These are early previews released in order to obtain real-world feedback during development. The canary channel will always have the latest build, including updates to beta or stable releases. We recommend running canary builds side-by-side with a beta or stable installation.
- **Dev:** Canary builds after a full round of internal testing get promoted into the Dev Channel.
- **Beta:** Release candidates based on stable canary builds, released and updated to obtain feedback prior to the stable release. The beta channel will be updated with new stable builds until a new canary build goes to beta.
- **Stable:** The official stable release, as available from the Android Developer site.

Download the full installations of the build offered in each of these channels: [Canary](#), [Dev](#), [Beta](#), [Stable](#).

Alternatively, you can build it yourself, following the instructions in [Build Overview](#).

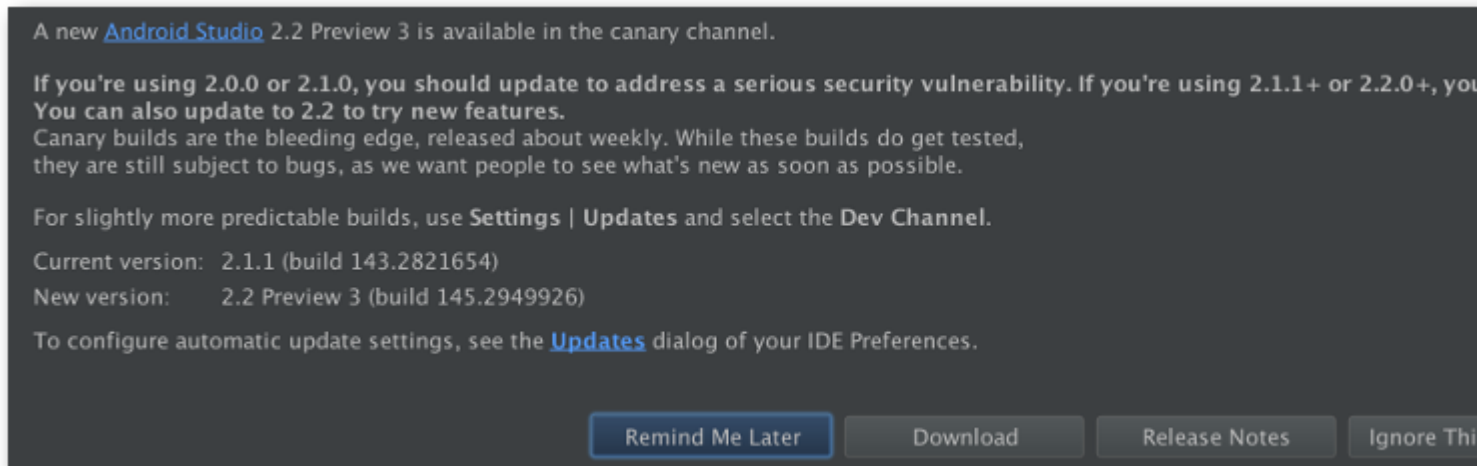
Selecting an Update Channel

To select the update channel for an Android Studio installation go through:

File > Settings > System Settings > Updates

and choose the appropriate channel on which to check for updates:

When an update is available, you'll be prompted by the IDE:



Selecting update will display the update information dialog with details information on the patch available, its size and its channel:

Read Android Studio updates online: <https://riptutorial.com/android-studio/topic/1998/android-studio-updates>

Credits

S. No	Chapters	Contributors
1	Getting started with android-studio	Ajay , Anurag , ARUN.G , Atif Farrukh , CJBS , Community , Daniel Nugent , H. Pauwelyn , Inzimam Tariq IT , kalan , murf , Nissim R , Paul Thompson , petrumo , piotrek1543 , pRaNaY , Rob R. , SaravInfern , Shashanth , sudo , Valentin Kuhn
2	Android Studio optimization	Kevin Cooper , manfcas , piotrek1543 , UmAnusorn , Yury Fedorov
3	Android Studio Tips and Tricks	Damindu , hellyale , meunicorn , piotrek1543 , Shashanth , the_new_mr
4	Android Studio updates	piotrek1543 , Yury Fedorov