



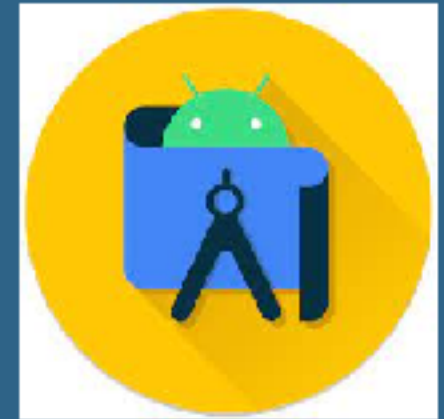
Migrate an Existing App to Compose





1. Install Android Studio Preview

- Beta: Artic Fox
 - Canary: (Bumblebee)
- Or





2. Add Gradle dependencies

```
implementation "androidx.compose.runtime:runtime:$composeVersion"  
implementation "androidx.compose.ui:ui:$composeVersion"  
implementation "androidx.compose.foundation:foundation:$composeVersion"  
implementation "androidx.compose.foundation:foundation-layout:$composeVersion"  
implementation "androidx.compose.material:material:$composeVersion"  
implementation "androidx.compose.runtime:runtime-livedata:$composeVersion"  
implementation "androidx.compose.ui:ui-tooling:$composeVersion"  
implementation "com.google.android.material:compose-theme-adapter:$composeVersion"  
implementation "androidx.activity:activity-compose:$composeVersion"
```

app module

Activity
Migration

1.0.0-rc01





3. Enable compose as build feature

```
android {  
  ...  
  buildFeatures {  
    ...  
    compose true  
  }  
  ...  
}
```

app module





Before Migration

Layout where the Custom ViewGroup is used

Custom
ViewGroup

ViewGroup
extended
<<Kotlin>>

layout_view_group
<<XML CODE>>

```
<LinearLayout>
```

```
...
```

```
<CustomViewGroup  
  android:id="@+id/customViewGroup1"  
  ... />
```

```
</LinearLayout>
```



After Migration



Activity Layout where the
Custom ViewGroup is used

**Compose
Function**

```
<LinearLayout>
  ...
  <androidx.compose.ui.platform.ComposeView
    android:id="@+id/customViewGroup1"
  ... />
</LinearLayout>
```

```
Activity {
  onCreate() {
    ...
    binding.customViewGroup1.setContent {
      ComposeFunction()
    }
  }
}
```



Activity Migration



```
<LinearLayout>  
  <androidx.compose.ui.platform.ComposeView  
    ... />  
  <androidx.compose.ui.platform.ComposeView  
    ... />  
  <androidx.compose.ui.platform.ComposeView  
    ... />  
</LinearLayout>
```

Hey XML,
I'll remove you from
my project



Activity Migration



```
Activity {  
  onCreate() {  
    ...  
    setContent {  
      Column {  
        ComposeFunction1()  
        ComposeFunction2()  
        ...  
      }  
    }  
  }  
}
```

**It is the same
of Linear
Layout with
vertical
orientation.**





How to know the equivalent UI Element in Compose?

The screenshot shows the JetpackCompose.app website. The header has links for 'COMPOSE CATALOG', 'WHICH COMPOSE API TO USE?', 'FAQ'S', and 'QUICK BITES'. The main content is split into two panels. The left panel, titled 'If you used...', shows 'LinearLayout' in a large serif font, with 'In Classic Android' and a robot icon below it. The right panel, titled 'Then, you could use...', shows 'Row/Column' in a large serif font, with 'In Jetpack Compose' and a rocket icon below it. At the bottom of the right panel, there are links for 'EXAMPLE TO HELP YOU GET STARTED' and 'OFFICIAL DOCS'.

JetpackCompose.app

COMPOSE CATALOG WHICH COMPOSE API TO USE? FAQ'S QUICK BITES

If you used...

LinearLayout

In Classic Android 🤖❤️

Then, you could use...

Row/Column

In Jetpack Compose 🚀

👉 EXAMPLE TO HELP YOU GET STARTED 👉

OFFICIAL DOCS



Tip: <https://www.jetpackcompose.app/>

