

# Creating Android Application Components

---

Within an Android application the following main components are distinguished:

- [Activities](#). An *activity* is a component that implements a screen to interact with the user.
- [Fragments](#). A *fragment* implements a part of a screen implemented by an *activity*.
- [Applications](#).
- [Services](#). A *service* is a component that runs in the background mode.
- [Broadcast Receivers](#) enable reception of [intents](#) from the system or other applications without running other activities or services of the application.
- [Android Remote Interface](#).

The creation procedure is identical for all types of components. Upon creation, a class that implements the component is created and the new component is automatically declared in the [AndroidManifest.xml](#) file.

On this page:

- [Creating an Android component](#)
- [Navigating between an activity or a fragment and its related layout definition file](#)
- [Navigating from a component to its declaration in the AndroidManifest.xml file](#)

## Creating an Android component

1. In the Project view, right-click the destination package where the application classes are stored.
2. On the context menu, click **New**, then choose **Android Component** on the submenu.
3. In the **New Android Component** dialog box that opens, specify the following:
  1. In the **Name** text box, type the name of the class that implements the component.
  2. From the **Kind** drop-down list, choose the desired component type:
    - Activity
    - Fragment
    - Application
    - Service
    - Broadcast Receiver
    - Android Remote Interface
  3. In the **Label** text box, type the component title to be displayed to the user.

- If no title is specified, the label assigned to the entire application will be displayed.
- No label is specified for a remote interface.

4. For an **Activity**, select the **Mark as startup activity** check box to have the activity displayed by default when the application starts.
5. For an **Activity** or a **Fragment**, select the **Create layout file** check box to have IntelliJ IDEA generate the stub of a related layout definition (content view). In the **Create Layout dialog box** that opens, specify the layout name, its root element, and the folder where the layout definition will be stored relative to the `res` folder.
  - If your application does not need to be compatible with various Android devices and therefore no **multiple screens** support is required, accept the default subfolder layout.
  - To provide **alternative resources**, specify the **resource qualifiers** that determine the specific device configuration. Move the relevant qualifiers from the **Available qualifiers** list to the **Chosen qualifiers** list and specify their values. IntelliJ IDEA appends all the selected qualifiers to the **Directory name** field with a dash character as separator.

For details, see [Creating Resources](#).

## Navigating between an activity or a fragment and its related layout definition file

You can jump from the source code of an activity or a fragment to the layout definition file which represents its content view and vice versa, from the layout definition to the source code.

- To jump from a component to its related layout definition, open the source code of the component, and do one of the following:
  - Click the  icon in the gutter area and choose the layout definition file in the **Go To Related Files** pop-up list.
  - On the main menu, choose **Navigate | Related File**.
- To jump from a layout definition to the source code of the corresponding component, open the layout definition file in the text mode and do one of the following:
  - Click the  icon in the gutter area.
  - On the main menu, choose **Navigate | Related File**.

## Navigating from a component to its declaration in the `AndroidManifest.xml` file

As soon as a component is created, it is immediately declared in the `AndroidManifest.xml` file. For components of the type *activity*, *fragment*, *service*, and *broadcast receiver*, you can jump to this declaration right from the component source code.

1. Open the source code of the component, and click the  icon in the gutter area.
2. If the component also has a related layout definition file, choose **AndroidManifest.xml** in the **Go To Related Files** pop-up list.

### See Also

#### Procedures:

- [Designing Layout of Android Application](#)

#### Reference:

- [New Android Component Dialog](#)
- [Android Reference](#)
- [Run/Debug Configuration: Android Application](#)
- [Run/Debug Configuration: Android Test](#)

#### Web Resources:

- [Developer Community](#) 