

Adding and Editing Layout Components Using Android UI Designer

A *layout* defines the user interface of an [activity](#) or an [app widget](#) (fragment). Layouts are declared in XML *resource definition files*. See the topic [Creating Resources](#) to find out how to create resource folders and resource definition files.

With IntelliJ IDEA, you can build the design of your application without editing the layout definition files manually and check how the application design is rendered in various target environments without running the application of any physical or even virtual devices.

Alternatively, edit the layout definition files manually, possibly using Android-specific refactoring provided by `%product`, and [preview the changes](#) that are immediately reflected in the dedicated [Preview](#) tool window, where you can adjust the layout to various platforms and devices. To switch to the manual mode, click the **Text** tab or choose **Go To Declaration** from the [context menu](#) in the **Design** pane or in the **Component Tree**.

Designing your layout in the visual mode is performed in the [Design](#) pane of the [Android UI Designer](#) tool window. The pane is located in the central part of the **UI Designer** (assuming the default tool window layout and visibility). When you open a layout definition file for editing, the pane appears in the editor tab by default. If you were editing the layout definition file manually and then switched to the visual mode by clicking the **Design** tab, the pane opens on the tab where the edited layout definition file is opened,

The pane shows a rectangular canvas that is synchronized with the current layout definition file and with the **Component Tree** view, so any changes to the canvas are reflected there accordingly. Accordingly, as soon as IntelliJ IDEA detects any discrepancy in the code generated according to the designed layout, a **Warnings** drop-down list  is added on the pane toolbar. Open the list to view the warnings. the number of warnings is shown on top.

To build the design of an Android application visually, perform these general steps:

- [Add predefined components from the Palette](#)
- [Add user-defined components and components from Android SDK](#)
- [Arrange the components](#)
- [Specify component properties](#)
- [Convert components into other types preserving the common properties](#)
- [Preview the layout](#)

To add a predefined component to the canvas, do one of the following

- Select the required element in the **Palette** pane and drag and drop it to the canvas in the **Design** pane.
- Click the required element in the **Palette** pane and then click the **Design** pane.

Every component added in either way is also added to the **Component Tree** and is declared in the layout definition file.

To add a component defined either in your project or in your project Android SDK

1. Expand the **Custom** area in the **Palette** pane.
2. Do one of the following:
 - To embed a layout, click the **Include** icon. In the **Resources** dialog box that opens, click the **Project** tab to search among your layout definitions or click the **System** tab to search in the SDK.
 - To add a combination of components, click **Fragment** and choose the fragment to embed in the **Resources** dialog box that opens.
 - To add a [user-defined view](#), click **Custom View:** and choose the view to insert in the dialog box that opens.

To find the proper position for a component

The canvas is synchronized with the **Component Tree**, so you can arrange components by moving them in either of these in either pane.

- Select the required component on the canvas or in the **Component Tree** and drag it to the right position or copy and paste it using the [context menu](#).

To specify component properties, perform these general steps

You can set the values for the mandatory properties of a component right in the canvas or switch to the **Properties** pane for a more detailed configuration of the component properties, possibly customizing the contents of the pane.

- To assign the values for the essential properties right in the canvas, click the component in question twice and specify the values in the pop-up dialog box that opens.
- To configure the component properties with more details, select the component in the canvas or in the **Component Tree**, switch to the **Properties** pane, and specify the values for the properties of your choice.
 - By default, the pane shows only a standard set of properties, and the most frequently used ones are displayed **bold**. To have the pane display all the properties that are defined for the selected component according to the specification, toggle the **Show expert properties** button  on the toolbar.
 - To view a brief documentation for the selected property from the [Android reference](#), click the **Show documentation** button  on the toolbar or press **Ctrl+Q** or **Alt+Button2 Click**.
 - Properties with updated values are highlighted blue. To remove the value you specified for a property and return to the default value, select the property and click the **Restore default value** button  on the toolbar.

To convert a component into a component of another type

In some cases you may need to transform an already fully configured component into a component of another type. With IntelliJ IDEA, you can do it without losing the specified properties: all the properties that are common for both types will be preserved in the new component. This operation is referred to as *morphing*.

1. Select the component to transform in the canvas or in the **Component Tree** and choose **Morphing** on the context menu of the selection.
2. IntelliJ IDEA shows a list of compatible component types, that is, the types into which the selected component can be converted. Choose the target type and [configure the properties](#) that were not configured in the original component.

See Also

Procedures:

- [Designing Layout of Android Application](#)
- [Previewing Output of Layout Definition Files](#)

Reference:

- [Designer Tool Window](#)

External Links:

- <http://blogs.jetbrains.com/idea/2012/06/android-ui-designer-coming-in-intellij-idea-12/> 

Web Resources:

- [Developer Community](#) 