

## Refactoring Android XML Layout Files

---

In addition to [common refactoring](#), IntelliJ IDEA provides a number of Android-specific refactorings for *Android layout definition XML files*. Most of these refactorings are available in both of the [design modes](#):

- In the manual mode, right in the editor tab where the target layout definition file is opened and from the [Structure tool window](#).
- In the [Android UI Designer](#), from the [Design pane](#) or [Component Tree](#).

You can toggle between these modes by switching between the **Design** and **Text** tabs in the editor where the layout definition file is opened.



In this topic:

- [Extract Style](#)
- [Inline Style](#)
- [Extract Layout](#)
- [Inline Layout Refactoring](#)

### Extract Style

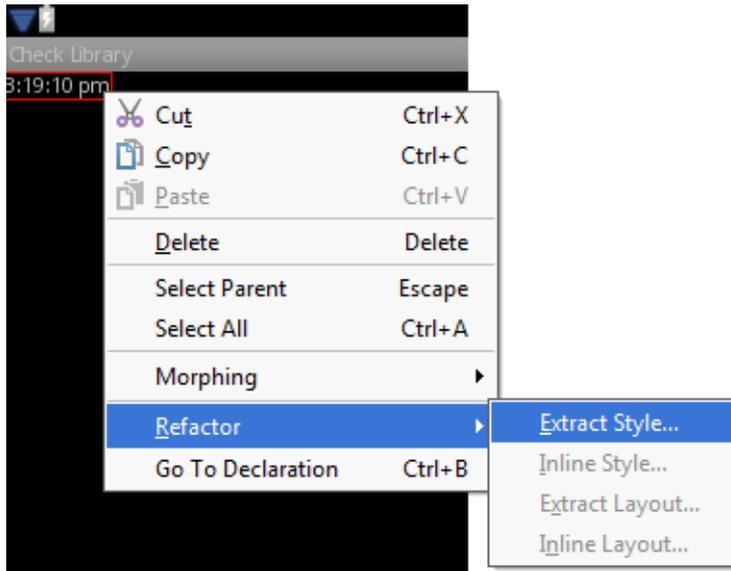
[Android styles](#) help separate the application design from the application functionality, just like [Cascading Style Sheets \(CSS\)](#).

With the *extract style* refactoring, you can create a style from a layout XML tag. IntelliJ IDEA creates a style declaration in the `/res/values/styles.xml` file and converts the tag attributes into style properties declared in the `item` elements. The benefit is that now to edit the layout you no longer need to parse the entire resource definition file updating the tag attributes. All you need is update the style definition whereupon the changes are applied automatically wherever the style is referenced.

## To apply the extract style refactoring

1. Do one of the following:
  - In the *visual mode*, select the component where the source XML tag is used in the definition. Select the component on the canvas in the **Design** pane or in the **Component Tree**.
  - In the *manual mode*, position the cursor anywhere inside the source XML tag.
2. On the main menu, choose **Refactor | Refactor This** or press **Ctrl+Shift+Alt+T**. In the pop-up list that appears, choose **Style** from the **Extract** group.

Alternatively, choose **Refactor | Extract | Extract Style** on the context menu of the selection.



3. In the **Extract Android Style** dialog box that opens, specify the name for the style to be created and appoint the tag attributes to be included as style properties by selecting the check boxes next to the names of relevant attributes.
4. To have IntelliJ IDEA automatically update the layout definition by replacing tag attributes with references to the newly created style, select the **Launch 'Use Style Where Possible' refactoring after the style is extracted** checkbox.

## Inline style

This refactoring is opposite to the *Extract Style* refactoring and results in adding all the attributes defined in a style to one or all components where this style is applied. The refactoring is also used when you need to merge a [parent style with its inheritor](#).

You can invoke the *Inline Style* refactoring from a style definition or from a component to which the style is applied. In the first case, IntelliJ IDEA removes the style definition and adds the corresponding attributes to all the components where it is used. When the refactoring is invoked from a component, you can apply it either to all the components with this style or only to the current one.

## To apply the Inline Style refactoring, do one of the following

- . To invoke the refactoring from a style definition:
  1. Open the style definition file, position the cursor at the name of the style to be inlined, and choose **Refactor | Inline** on the context menu or press **Ctrl+Alt+N**.
  2. The **Inline Android Style** dialog box that opens, provides only one option **Inline all references and remove the style** which is chosen by default. As a result, IntelliJ IDEA converts the attributes of the style into XML tags, adds them to the definitions of all the components where the style is used, and removes the style definition
    - To apply the changes immediately, click **Refactor**.
    - To open the **Find tool window** and **preview the changes before applying**, click **Preview**
- To invoke the refactoring from a component definition:
  1. Select a component with the style in question:
    - In the *>visual mode*, select a component with this style in the **Design** pane or in the **Component Tree**. Then choose **Refactor | Inline Style** on the main menu or on the context menu of the selection.
    - In the *manual mode*, select a component with this style in the **Structure** view and choose **Refactor | Inline Style** on the main menu or on the context menu of the selection.
  2. In the **Inline Android Style** dialog box that opens, configure the refactoring:
    - To have IntelliJ IDEA convert the attributes of the style into XML tags, add them to the definitions of all the components where the style is used, and remove the style definition, choose **Inline all references and remove the style**.
    - To have the refactoring applied to the current component only, choose **Inline this usage and keep the style**.
  3. Specify how you want to apply the changes.
    - To apply the changes immediately, click **Refactor**.
    - To open the **Find tool window** and **preview the changes before applying**, click **Preview**

## Extract Layout

IntelliJ IDEA supports [re-using Android layouts](#): IntelliJ IDEA moves the a part of an existing layout definition to a separate layout definition file and references it in the original layout definition through an automatically inserted `<include/>` tag. In IntelliJ IDEA, this operation is called *extract layout refactoring*.

## To extract a layout

1. Select the fragment to extract as a separate layout. Do one of the following:
  - In the *manual mode*, select the code fragment to extract or position the cursor in the root element of the fragment.
  - In the *visual mode*, select the component or a group of components to be extracted in the **Design** pane or in the **Component Tree**.
2. On the main menu, choose **Refactor | Refactor This** or press **Ctrl+Shift+Alt+T**. In the pop-up list that appears, choose **Layout** from the **Extract** group.

Alternatively, choose **Refactor | Extract | Extract Layout** on the context menu of the selection.

3. In the **Extract Android Layout** dialog box that opens, specify the name of the file where the extracted layout definition will be stored and the name of its parent folder. You do not need to create the file and the folder in advance, IntelliJ IDEA will create them itself.
4. Create a list of configuration qualifiers for the layout. This list determines the set of *alternative resources* that should be provided to enable running the application in specific target device configurations.
  - To add a qualifier to the set, select it in the **Available Qualifiers** list, click the *chevron* button **»**, and specify the value of the qualifier.
  - To remove a qualifier from the set, select it in the **Chosen Qualifiers** list and click the *chevron* button **«**.

For detailed information on the qualifiers, their meaning, acceptable values and format, see [specification of configuration qualifiers for alternative resources](#).

## Inline Layout Refactoring

This refactoring is opposite to the *Extract Layout* refactoring and results in embedding a layout that was referenced through an `<include/>` tag and removing the tag.

The *Inline Layout* refactoring is invoked from the `<include/>` tag in the layout definition file. You can have IntelliJ IDEA apply it either to all the references and remove the layout definition file or only to the current reference.

## To inline a layout

1. Open the XML definition file from which the layout in question is referenced.
2. Navigate to the `<include/>` tag which references the layout in question and position the cursor at the layout name.
3. Do one of the following:
  - Choose **Refactor | Inline** on the main menu.
  - Choose **Refactor | Refactor This** on the main menu, and then choose **Inline** in the pop-up list that opens,
  - Press **Ctrl+Alt+N**.
  - Choose **Refactor | Inline** on the context menu.
4. In the **Inline Android Layout** dialog box that opens, configure the refactoring:
  - To have IntelliJ IDEA embed the layout into all files where it is referenced through `<include/>` tags and remove the referenced layout definition file, choose **Inline all references and remove the file**.
  - To have the refactoring applied to the current reference only, **Inline this usage and keep the file**.
5. Specify how you want to apply the changes.
  - To apply the changes immediately, click **Refactor**.
  - To open the [Find tool window](#) and [preview the changes before applying](#), click **Preview**

## See Also

### Procedures:

- [Adding and Editing Layout Components Using Android UI Designer](#)
- [Managing Resources](#)
- [Previewing Output of Layout Definition Files](#)

### Reference:

- [Android Layout Preview Tool Window](#)
- [Inline Android Style Dialog](#)

### External Links:

- <http://blogs.jetbrains.com/idea/2012/09/new-xml-refactorings-for-android-developers/> 

### Web Resources:

- [Developer Community](#) 