

Implement Intention

```
override fun invoke(project: Project, editor: Editor?, file: PsiFile?) {  
    val rootTag = file.getRootTag() ?: return  
    val xmlnsAttribute = rootTag.attributes  
        .filter { it.name.startsWith("xmlns:") }  
    val attributeText = xmlnsAttribute  
        .map { it.text }  
        .joinToString(separator = " ")  
    val factory = XmlElementFactory.getInstance(project)  
  
    xmlnsAttribute.forEach { it.delete() }  
    val layoutXmlTag = factory.createTagFromText("<layout  
$attributeText>${rootTag.text}</layout>", XMLLanguage.INSTANCE)  
    rootTag.replace(layoutXmlTag)  
}
```

<LinearLayout

```
    xmlns:android="http://schemas.android.com/apk/res/android"  
    xmlns:tools="http://schemas.android.com/tools"  
    xmlns:app="http://schemas.android.com/apk/res-auto"  
    android:orientation="vertical"  
    android:layout_width="match_parent"  
    android:layout_height="match_parent">
```

</LinearLayout>

Implement Intention

```
override fun invoke(project: Project, editor: Editor?, file: PsiFile?) {  
    val rootTag = file.getRootTag() ?: return  
    val xmlnsAttribute = rootTag.attributes  
        .filter { it.name.startsWith("xmlns:") }  
    val attributeText = xmlnsAttribute  
        .map { it.text }  
        .joinToString(separator = " ")  
    val factory = XmlElementFactory.getInstance(project)  
  
    xmlnsAttribute.forEach { it.delete() }  
    val layoutXmlTag = factory.createTagFromText("<layout  
$attributeText>${rootTag.text}</layout>", XMLLanguage.INSTANCE)  
    rootTag.replace(layoutXmlTag)  
}
```

```
xmlns:android="http://schemas.android.com/apk/res/android" xmlns:tools="http://  
schemas.android.com/tools" xmlns:app="http://schemas.android.com/apk/res-auto"
```