

# Generify Refactoring

The **Generify** refactoring is designed to transform existing code that does not use Generics, into the Generics-aware code. The refactoring analyzes existing code, and for each raw type creates safe and consistent parameter type.

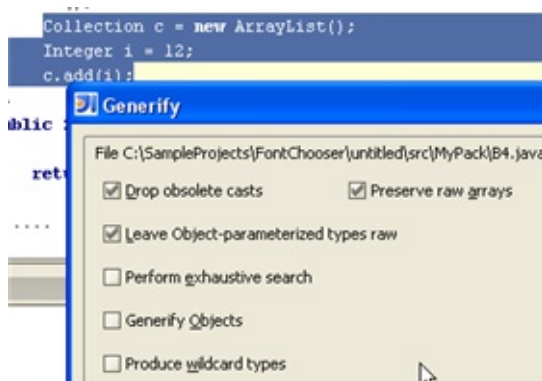
IntelliJ IDEA tries to generate code, which is as correct as possible from the Java point of view. In other words, each context introduces some type restrictions, and the refactoring produces the best possible type, that does not contradict to the existing contexts.

## Example

Before	After
<pre>public void method() {     List list = new LinkedList();     list.add("string"); }</pre>	<pre>public void method() {     List&lt;String&gt; list = new LinkedList&lt;String&gt;();     list.add("string"); }</pre>

## To generify

1. Select the level of code transformation, which can be a method, a class, a package or directory, in the Project or Structure view, or place the cursor on the class or method name in the editor. If you want to apply generics to a single code fragment, just select it in the editor.
2. On the main menu, or on the context menu of the selection, choose **Refactor | Generify**. The **Generify** dialog box appears.



3. Define the refactoring options. Refer to the [dialog description](#) for details.
4. [Preview and apply changes](#).

```
Collection<Integer> c = new ArrayList<Integer>();
Integer i = 12;
c.add(i);
```

## See Also

Reference:

- [Generify Dialog](#)

Web Resources:

- [Developer Community](#)