

## 1 How to add a new code generator

1. Add a new override option `OVERRIDE_NEW_CLASS` to `concretesyntax.ecore`. The easiest way to do this, is to open `concretesyntax.ecore` with the Eclipse text editor and add a line like `<eLiterals name="OVERRIDE_NEW_CLASS" value="0" literal="overrideNewClass"/>` to the literal list of enumeration `OptionTypes`. Instead of 0 for attribute value, use the next free integer instead.
2. Reload `concretesyntax.genmodel`.
3. Regenerate the model code from `concretesyntax.genmodel`.
4. Create a new generator class called `NewClassGenerator.java`, which extends either `JavaBaseGenerator` or `UIJavaBaseGenerator`. Place the new generator class in the respective package in plug-in `org.emftext.sdk.codegen.resource` or `org.emftext.sdk.codegen.resource.ui`.
5. Add field `newClassClassName` to `ResourceBaseGenerator` or `UIResourceBaseGenerator`.
6. Add constant `NEW_CLASS` to `TextResourceArtifacts` or `TextResourceUIArtifacts` using a line like: 

```
public final static ArtifactDescriptor<GenerationContext,
ArtifactParameter<GenerationContext>> NEW_CLASS = new ArtifactDescriptor<GenerationContext,
ArtifactParameter<GenerationContext>>(XYZ_PACKAGE, "", "NewClass", NewClassGenerator
OptionTypes.OVERRIDE_NEW_CLASS);
```
7. Initialize field `newClassClassName` in method `initilizeClassNames()` `ResourceBaseGenerator` or `UIResourceBaseGenerator`.
8. Add line `add(creators, TextResourceArtifacts.NEW_CLASS);` to class `ResourcePluginContentCreator` or line `add(creators, TextResourceUIArtifacts.NEW_CLASS);` to class `ResourceUIPluginContentCreator`.
9. Implement `NewClassGenerator`. Put constants for required classes into `IClassNameConstants` or `IUIClassNameConstants`.
10. Add documentation for the new override option `OVERRIDE_NEW_CLASS` to `concretesyntax.emfdoc` and regenerated the CS model code. Make sure you've got EMFDoc installed. Otherwise the documentation will not be added to `concretesyntax.ecore`.