# Code Styles



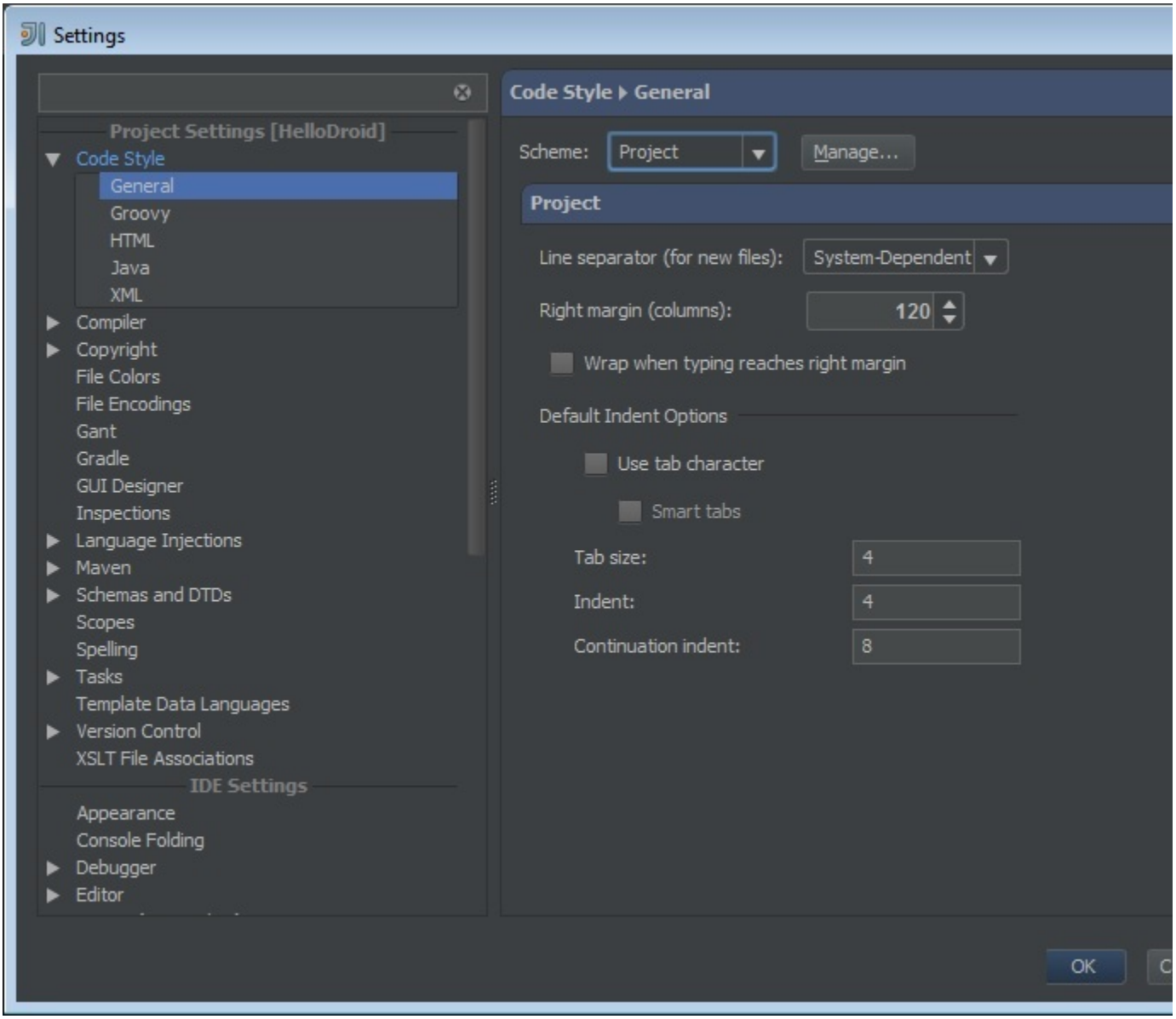


Developers who have used Eclipse for years may have built some strong expectations as to how an IDE has to behave and especially format code. In particular, you may be used to having the IDE to save files automatically or to have tabs used instead of spaces or extra blank lines removed before a closing bracket.

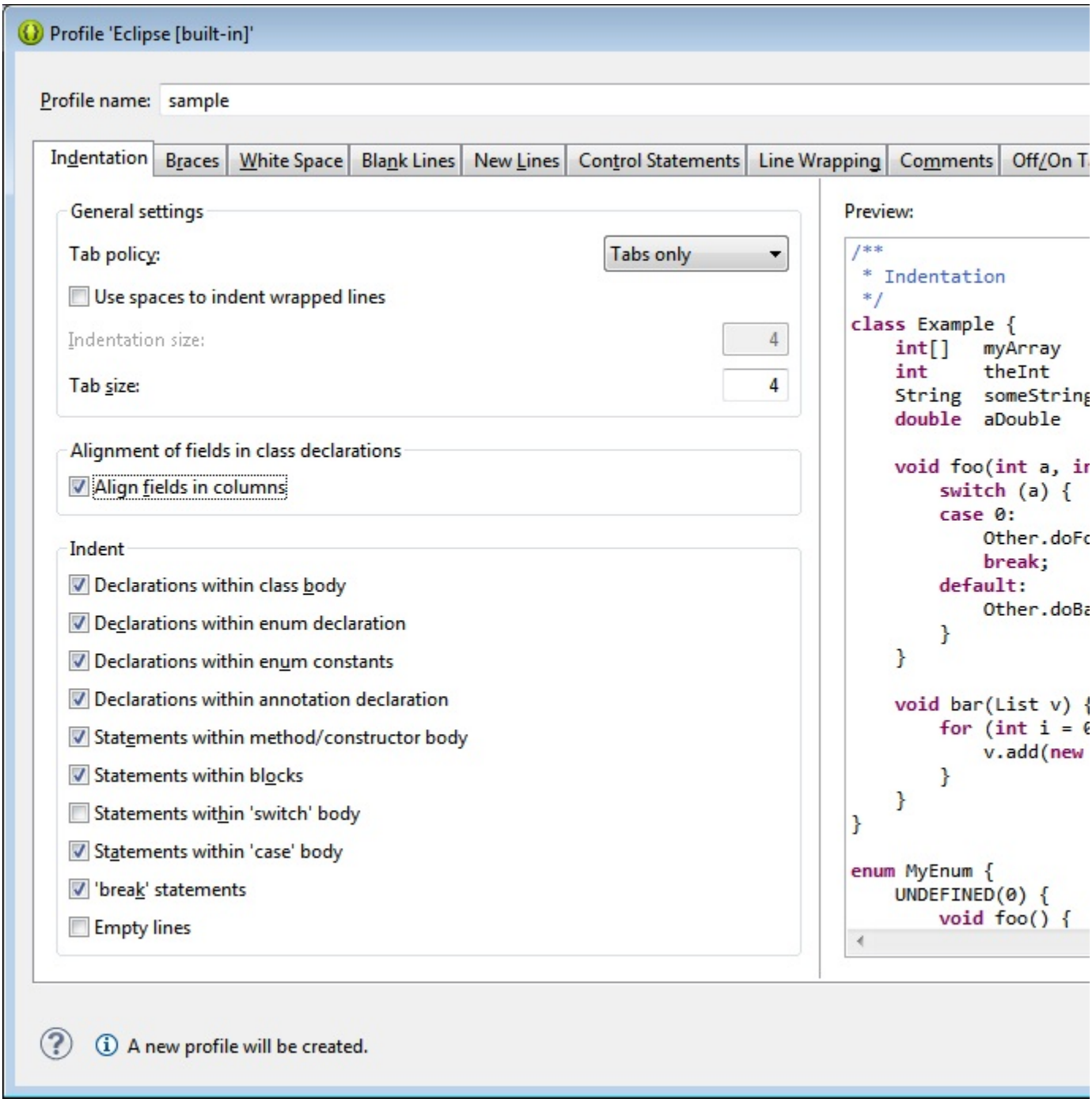
A different set of coding styles may obviously create trouble in teams where not all members use the same IDE. Let's review a few strategies to share coding styles between IntelliJ IDEA and Eclipse.

## Supported Code Styles

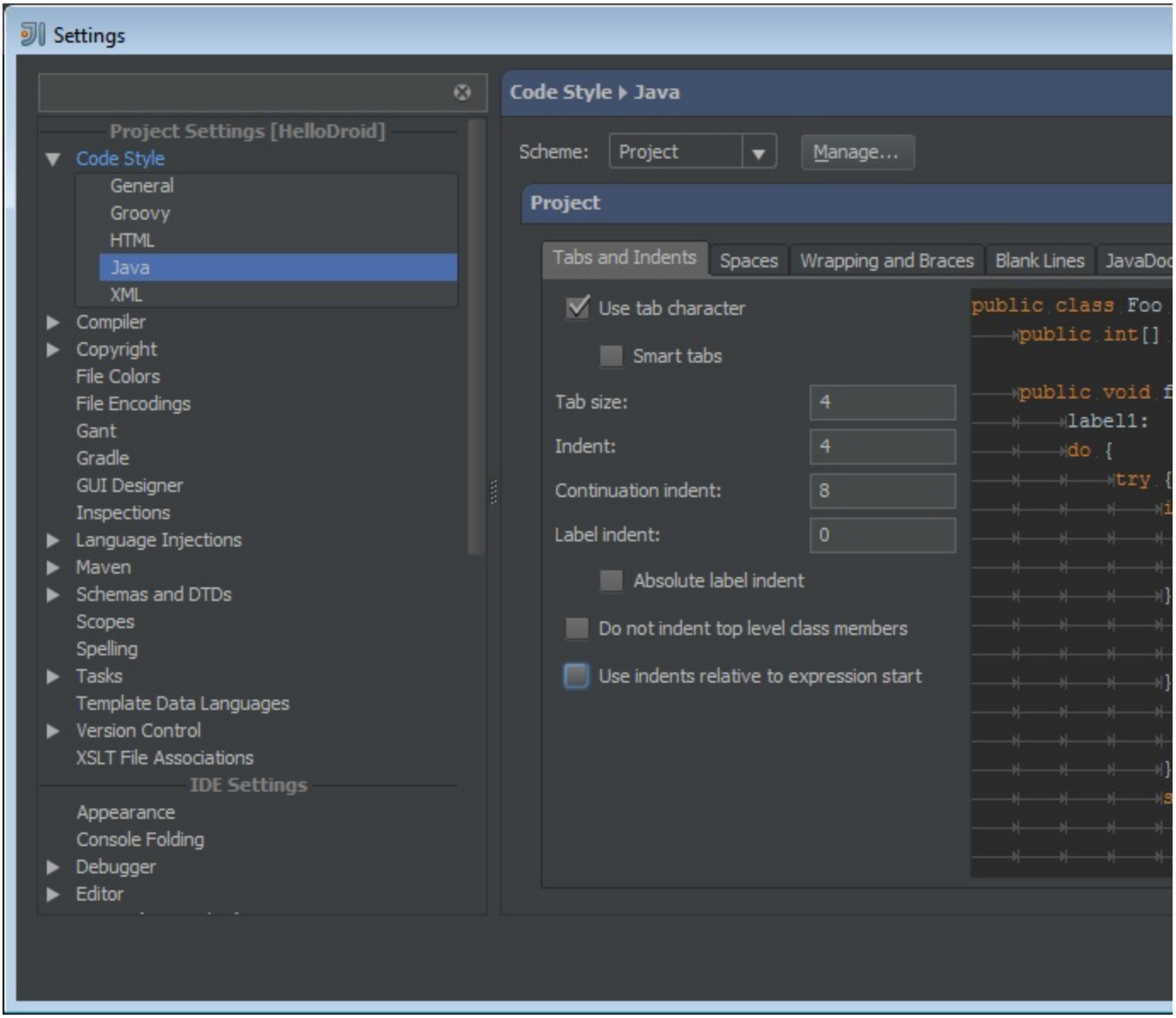
Although through different user interfaces, Eclipse and IntelliJ IDEA offer nearly the same set of coding styles for you to configure. Configuration takes place from theFile  Settingsdialog box after selecting the Code Style node. You have styles grouped in various categories: General, Java, Groovy, HTML and XML.



Of particular relevance are the code style settings available for the Java language. Settings are grouped in different categories in Eclipse and IntelliJ IDEA. However as the next two screenshots show, categories address nearly the same set of parameters.



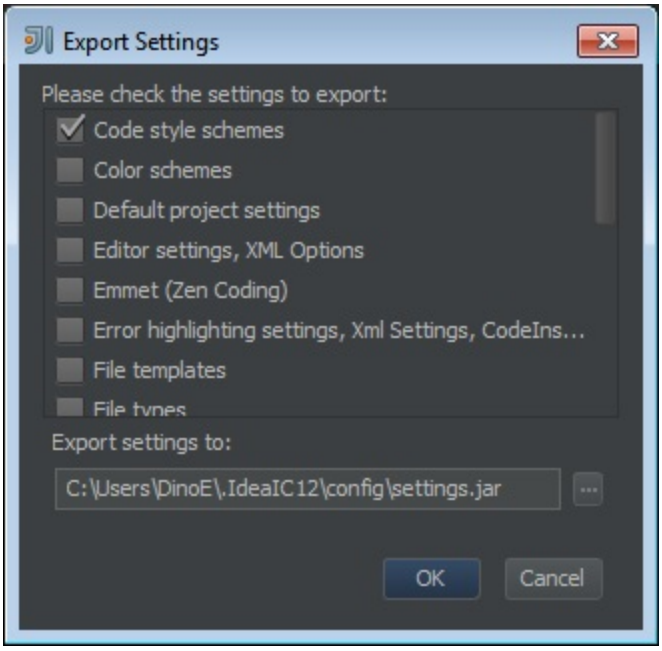

Eclipse seems to have a few more tabs compared to IntelliJ, but IntelliJ fits more settings in each tab.




Both Eclipse and IntelliJ IDEA offer a preview of the currently selected code styles, at least those that impact directly on code writing.

## Sharing IntelliJ IDEA Code Style Settings

Both Eclipse and IntelliJ IDEA let you save all settings to a file. A settings file can then be imported later into another instance of the IDE. Note, though, that Eclipse has a direct Export button to export code styles. In IntelliJ IDEA, instead, you export and import all IDE settings, including code style schemes. You selectFile  Export Settingsand select Code style schemes.




To import a previously saved settings file you selectFile  Import Settings.
IntelliJ IDEA allows to export and import IDE settings so that all users on a team can share the same settings, but this doesn't solve the problem of teaching IntelliJ IDEA to apply the settings in use in Eclipse.

## Applying Eclipse Code Styles to IntelliJ IDEA

There are two approaches to import the Eclipse code styles in IntelliJ IDEA.

1. A developer on the team takes the burden of configuring his own instance of IntelliJ IDEA based on whatever collection of settings the team agrees on--including the common settings of Eclipse.
2. You can install the following IntelliJ IDEA plug-in: http://plugins.jetbrains.com/plugin/?id=6546. The plug-in lets you use the Eclipse's code formatter directly from IntelliJ. In doing so, it solves two problems at the same time. First, it lets developers to work in IntelliJ IDEA with the same code style they were used to in Eclipse. Second, it helps maintaining a common coding style in collaborative environments where both IntelliJ IDEA and Eclipse are used.

Because at the end of the day both IDE save current settings to a local XML file, a smart tool can read one file and convert it to another schema automatically to the extent that it is possible. This is precisely what the aforementioned plug-in does: it reads the settings of a given Eclipse project and imports those settings into the IntelliJ IDEA local file storing settings.

## Shared IDE Strategy

The aforementioned plug-in brings Eclipse settings into IntelliJ IDEA. In case of a team where some members use IntelliJ IDEA and some others use Eclipse, the easiest is bringing Eclipse settings into IntelliJ through the plug-in.

Any other approach would anyway require that all relevant settings are manually entered and kept synced in both IDEs. It is then up to each team to decide about which code style settings are relevant to keep synced.

Previous Top Next