

# IDEA Code formatting subsystem



Redirection Notice

This page will redirect to [http://www.jetbrains.org/intellij/sdk/docs/reference\\_guide/custom\\_language\\_support/code\\_formatting.html](http://www.jetbrains.org/intellij/sdk/docs/reference_guide/custom_language_support/code_formatting.html).

## Purpose

The purpose of this document is to provide information about general design, API and further evolution of IDEA code formatting subsystem

## Table of Contents

- [Redirection Notice Icon This page will redirect to http://www.jetbrains.org/intellij/sdk/docs/reference\\_guide/custom\\_language\\_support/code\\_formatting.html.](#)
- [Purpose](#)
- [Table of Contents](#)
- [Design](#)
  - [Overview](#)
  - [Block API](#)
  - [Formatting subsystem API](#)
- [Evolution](#)
  - [Profiling](#)
  - [Java & XML mix](#)

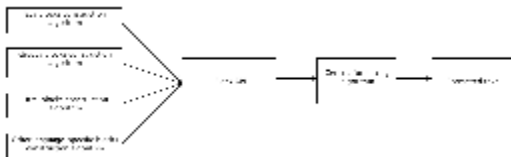
## Design

### Overview

The general idea of code formatting approach used by IntelliJ platform is to have the following:

- language-agnostic API for representing target source file to format as a structure of visual blocks;
- language-agnostic API that allows to define formatting rules for visual blocks mentioned above;
- generic formatting algorithm that applies target formatting options to visual blocks mentioned above;
- language-specific algorithms that build visual blocks from source code;

Here is a diagram that shows the parts mentioned above:



### Block API

'Block' is an object that IS-A `com.intellij.formatting.Block`. It has the following responsibilities:

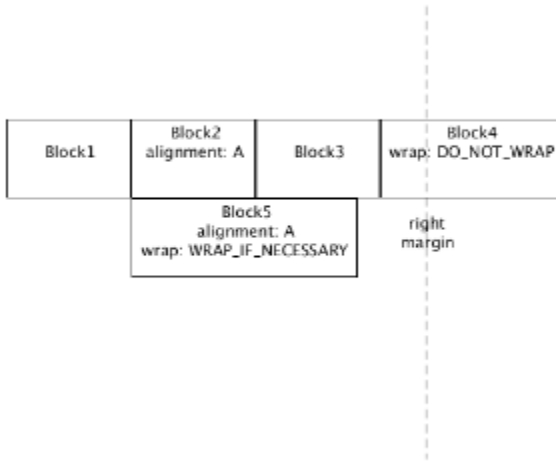
- encapsulates target text range;
- can aggregate sub-blocks;
- can aggregate target formatting options;

Formatting options

All options are created using static factory methods on corresponding classes (e.g. 'Indent' option is created via various 'Indent .getXxxIndent()' methods).

- Wrap - defines if the line should be wrapped before the block if it exceeds right margin;
- Indent - defines how the block is indented relative to its parent block;
- Spacing - encapsulates information about white spaces and line feeds to use between this block and its previous block;
- Alignment - all blocks that share the same alignment object are aligned together;

Example



Here 'Block4' is not moved to the new line because its 'wrap' property is set to 'DO\_NOT\_WRAP' and 'Block5' is aligned to 'Block2' because the same Alignment object was set to their 'alignment' property.

## Formatting subsystem API

### Evolution

### Profiling

It looks like formatting subsystem contains many potentially expensive operations (e.g. converting whole document text to string in order to perform 'equals()' comparison at 'com.intellij.psi.formatter.FormattingDocumentModelImpl.createOn()'). Also it's not quite clear why so many nested blocks are constructed for java source. That leads to intention to profile formatting subsystem processing and revise it as necessary.

### Java & XML mix

'CDATA' blocks processing is hard-coded to generic formatting logic - 'com.intellij.psi.formatter.DocumentBasedFormattingModel.replaceWhiteSpace()'. Looks rather suspicious and controversial to overall formatting subsystem design.