Package org.eclipse.jdt.core.formatter

Class DefaultCodeFormatterConstants

java.lang.Object
  org.eclipse.jdt.core.formatter.DefaultCodeFormatterConstants

```
public class DefaultCodeFormatterConstants
extends Object
```

Constants used to set up the options of the code formatter.

Since:

3.0

Restriction:

This class is not intended to be subclassed by clients.

Restriction:

This class is not intended to be instantiated by clients.

Field Summary

Fields

Modifier and Type	Field	Description
static final String	COMMON_LINES	FORMATTER / Value to set opening and closing parentheses location in common lines with their contents (or simply a single line if the parentheses are empty).
static final String	END_OF_LINE	FORMATTER / Value to set a brace location at the end of a line.
static final String	FALSE	FORMATTER / Value to set an option to false.
static final String	FORMATTER_ALIGN_ARROWS_IN_SWITCH_ON_COLUMNS	FORMATTER / Option to align arrows in switch on column - option id: "org.eclipse.jdt.core.formatter.align_arrows_in_switch_on_columns" - possible values: { TRUE, FALSE } - default: FALSE
static final String	FORMATTER_ALIGN_ASSIGNMENT_STATEMENTS_ON_COLUMNS	FORMATTER / Option to align assignment statements on column - option id: "org.eclipse.jdt.core.formatter.align_assignment_statements_on_columns" - possible values: { TRUE, FALSE } - default: FALSE
static final String	FORMATTER_ALIGN_FIELDS_GROUPING_BLANK_LINES	FORMATTER / Option to affect aligning on columns: groups of items are aligned independently if they are separated by at least the selected number of blank lines.
static final String	FORMATTER_ALIGN_SELECTOR_IN_METHOD_INVOCATION_ON_EXPRESSION_FI	FORMATTER / Option to indent method invocation chains based on the first line of the base expression rather than the last line
static final String	FORMATTER_ALIGN_TYPE_MEMBERS_ON_COLUMNS	FORMATTER / Option to align type members of a type declaration on column - option id: "org.eclipse.jdt.core.formatter.align_type_members_on_columns" - possible values: { TRUE, FALSE } - default: FALSE
static final String	FORMATTER_ALIGN_VARIABLE_DECLARATIONS_ON_COLUMNS	FORMATTER / Option to align variable declarations on column - option id: "org.eclipse.jdt.core.formatter.align_variable_declarations_on_columns" - possible values: { TRUE, FALSE } - default: FALSE
static final String	FORMATTER_ALIGN_WITH_SPACES	FORMATTER / Option to use spaces when aligning members, independent of selected tabulation character - option id: "org.eclipse.jdt.core.formatter.align_with_spaces" - possible values: { TRUE, FALSE } - default: FALSE
static final String	FORMATTER_ALIGNMENT_FOR_ADDITIVE_OPERATOR	FORMATTER / Option for alignment of expressions with additive operators (+, -) - option id: "org.eclipse.jdt.core.formatter.alignment_for_additive_operator" - possible values: values returned by `createAlignmentValue(boolean, int, int)` call - default: createAlignmentValue(false, WRAP_COMPACT, INDENT_DEFAULT)
static final String	FORMATTER_ALIGNMENT_FOR_ANNOTATIONS_ON_ENUM_CONSTANT	FORMATTER / Option for alignment of annotations on enum constant declaration - option id: "org.eclipse.jdt.core.formatter.alignment_for_annotations_on_enum_consta - possible values: values returned by `createAlignmentValue(boolean, int)` call - default: createAlignmentValue(true, WRAP_ONE_PER_LINE)

static final String	FORMATTER_ALIGNMENT_FOR_ANNOTATIONS_ON_FIELD	FORMATTER / Option for alignment of annotations on field declaration - option id: "org.eclipse.jdt.core.formatter.alignment_for_annotations_on_field" - possible values: values returned by `createAlignmentValue(boolean, int)` call - default: createAlignmentValue(true, WRAP_ONE_PER_LINE)
static final String	FORMATTER_ALIGNMENT_FOR_ANNOTATIONS_ON_LOCAL_VARIABLE	FORMATTER / Option for alignment of annotations on local variable - option id: "org.eclipse.jdt.core.formatter.alignment_for_annotations_on_local_variabl - possible values: values returned by `createAlignmentValue(boolean, int)` call - default: createAlignmentValue(true, WRAP_ONE_PER_LINE)
static final String	FORMATTER_ALIGNMENT_FOR_ANNOTATIONS_ON_METHOD	FORMATTER / Option for alignment of annotations on method declaration - option id: "org.eclipse.jdt.core.formatter.alignment_for_annotations_on_method" - possible values: values returned by `createAlignmentValue(boolean, int)` call - default: createAlignmentValue(true, WRAP_ONE_PER_LINE)
static final String	FORMATTER_ALIGNMENT_FOR_ANNOTATIONS_ON_PACKAGE	FORMATTER / Option for alignment of annotations on package declaration - option id: "org.eclipse.jdt.core.formatter.alignment_for_annotations_on_package" - possible values: values returned by `createAlignmentValue(boolean, int)` call - default: createAlignmentValue(true, WRAP_ONE_PER_LINE)
static final String	FORMATTER_ALIGNMENT_FOR_ANNOTATIONS_ON_PARAMETER	FORMATTER / Option for alignment of annotations on parameter - option id: "org.eclipse.jdt.core.formatter.alignment_for_annotations_on_parameter" - possible values: values returned by `createAlignmentValue(boolean, int)` call - default: createAlignmentValue(false, WRAP_NO_SPLIT)
static final String	FORMATTER_ALIGNMENT_FOR_ANNOTATIONS_ON_TYPE	FORMATTER / Option for alignment of annotations on type declaration - option id: "org.eclipse.jdt.core.formatter.alignment_for_annotations_on_type" - possible values: values returned by `createAlignmentValue(boolean, int)` call - default: createAlignmentValue(true, WRAP_ONE_PER_LINE)
static final String	FORMATTER_ALIGNMENT_FOR_ARGUMENTS_IN_ALLOCATION_EXPRESSION	FORMATTER / Option for alignment of arguments in allocation expression - option id: "org.eclipse.jdt.core.formatter.alignment_for_arguments_in_allocation_exp - possible values: values returned by `createAlignmentValue(boolean, int, int)` call - default: createAlignmentValue(false, WRAP_COMPACT, INDENT_DEFAULT)
static final String	FORMATTER_ALIGNMENT_FOR_ARGUMENTS_IN_ANNOTATION	FORMATTER / Option for alignment of arguments in annotation - option id: "org.eclipse.jdt.core.formatter.alignment_for_arguments_in_annotation" - possible values: values returned by `createAlignmentValue(boolean, int, int)` call - default: createAlignmentValue(false, WRAP_NO_SPLIT, INDENT_DEFAULT)
static final String	FORMATTER_ALIGNMENT_FOR_ARGUMENTS_IN_ENUM_CONSTANT	FORMATTER / Option for alignment of arguments in enum constant - option id: "org.eclipse.jdt.core.formatter.alignment_for_arguments_in_enum_constan - possible values: values returned by `createAlignmentValue(boolean, int, int)` call - default: createAlignmentValue(false, WRAP_COMPACT, INDENT_DEFAULT)
static final String	FORMATTER_ALIGNMENT_FOR_ARGUMENTS_IN_EXPLICIT_CONSTRUCTOR_CALL	FORMATTER / Option for alignment of arguments in explicit constructor call - option id: "org.eclipse.jdt.core.formatter.alignment_for_arguments_in_explicit_constr - possible values: values returned by `createAlignmentValue(boolean, int, int)` call - default: createAlignmentValue(false, WRAP_COMPACT, INDENT_DEFAULT)
static final String	FORMATTER_ALIGNMENT_FOR_ARGUMENTS_IN_METHOD_INVOCATION	FORMATTER / Option for alignment of arguments in method invocation - option id: "org.eclipse.jdt.core.formatter.alignment_for_arguments_in_method_invoc - possible values: values returned by `createAlignmentValue(boolean, int, int)` call - default: createAlignmentValue(false, WRAP_COMPACT, INDENT_DEFAULT)
static final String	FORMATTER_ALIGNMENT_FOR_ARGUMENTS_IN_QUALIFIED_ALLOCATION_EXPR	FORMATTER / Option for alignment of arguments in qualified allocation expression - option id: "org.eclipse.jdt.core.formatter.alignment_for_arguments_in_qualified_alloc - possible values: values returned by `createAlignmentValue(boolean, int, int)` call - default: createAlignmentValue(false, WRAP_COMPACT, INDENT_DEFAULT)

static final String	FORMATTER_ALIGNMENT_FOR_ASSERTION_MESSAGE	FORMATTER / Option for alignment of assertion message separator (:) - option id: "org.eclipse.jdt.core.formatter.alignment_for_assertion_message" - possible values: values returned by `createAlignmentValue(boolean, int, int)` call - default: createAlignmentValue(false, WRAP_NO_SPLIT, INDENT_DEFAULT)
static final String	FORMATTER_ALIGNMENT_FOR_ASSIGNMENT	FORMATTER / Option for alignment of assignment (=, +=, -=, *=, /=, %=, &=, ^=, |=, <<=, >>=, >>>=) - option id: "org.eclipse.jdt.core.formatter.alignment_for_assignment" - possible values: values returned by `createAlignmentValue(boolean, int, int)` call - default: createAlignmentValue(false, WRAP_NO_SPLIT, INDENT_DEFAULT)
static final String	FORMATTER_ALIGNMENT_FOR_BINARY_EXPRESSION	**Deprecated.** Use new settings instead: FORMATTER_ALIGNMENT_FOR_MULTIPLICATIVE_OPERATOR, FORMATTER_ALIGNMENT_FOR_ADDITIVE_OPERATOR, FORMATTER_ALIGNMENT_FOR_STRING_CONCATENATION, FORMATTER_ALIGNMENT_FOR_BITWISE_OPERATOR, FORMATTER_ALIGNMENT_FOR_LOGICAL_OPERATOR
static final String	FORMATTER_ALIGNMENT_FOR_BITWISE_OPERATOR	FORMATTER / Option for alignment of expressions with bitwise operators (&, ^, |) - option id: "org.eclipse.jdt.core.formatter.alignment_for_bitwise_operator" - possible values: values returned by `createAlignmentValue(boolean, int, int)` call - default: createAlignmentValue(false, WRAP_COMPACT, INDENT_DEFAULT)
static final String	FORMATTER_ALIGNMENT_FOR_COMPACT_IF	FORMATTER / Option for alignment of compact if - option id: "org.eclipse.jdt.core.formatter.alignment_for_compact_if" - possible values: values returned by `createAlignmentValue(boolean, int, int)` call - default: createAlignmentValue(false, WRAP_ONE_PER_LINE, INDENT_BY_ONE)
static final String	FORMATTER_ALIGNMENT_FOR_COMPACT_LOOP	FORMATTER / Option for alignment of compact loops - option id: "org.eclipse.jdt.core.formatter.alignment_for_compact_loops" - possible values: values returned by `createAlignmentValue(boolean, int, int)` call - default: createAlignmentValue(false, WRAP_ONE_PER_LINE, INDENT_BY_ONE)
static final String	FORMATTER_ALIGNMENT_FOR_CONDITIONAL_EXPRESSION	FORMATTER / Option for alignment of conditional expression - option id: "org.eclipse.jdt.core.formatter.alignment_for_conditional_expression" - possible values: values returned by `createAlignmentValue(boolean, int, int)` call - default: createAlignmentValue(false, WRAP_ONE_PER_LINE, INDENT_DEFAULT)
static final String	FORMATTER_ALIGNMENT_FOR_CONDITIONAL_EXPRESSION_CHAIN	FORMATTER / Option for alignment of conditional expression chains.
static final String	FORMATTER_ALIGNMENT_FOR_ENUM_CONSTANTS	FORMATTER / Option for alignment of enum constants - option id: "org.eclipse.jdt.core.formatter.alignment_for_enum_constants" - possible values: values returned by `createAlignmentValue(boolean, int, int)` call - default: createAlignmentValue(false, WRAP_NO_SPLIT, INDENT_DEFAULT)
static final String	FORMATTER_ALIGNMENT_FOR_EXPRESSIONS_IN_ARRAY_INITIALIZER	FORMATTER / Option for alignment of expressions in array initializer - option id: "org.eclipse.jdt.core.formatter.alignment_for_expressions_in_array_initializ - possible values: values returned by `createAlignmentValue(boolean, int, int)` call - default: createAlignmentValue(false, WRAP_COMPACT, INDENT_DEFAULT)
static final String	FORMATTER_ALIGNMENT_FOR_EXPRESSIONS_IN_FOR_LOOP_HEADER	FORMATTER / Option for alignment of initialization, termination, and increment expressions in 'for' loop header - option id: "org.eclipse.jdt.core.formatter.alignment_for_for_loop_header" - possible values: values returned by `createAlignmentValue(boolean, int, int)` call - default: createAlignmentValue(false, WRAP_NO_SPLIT, INDENT_DEFAULT)
static final String	FORMATTER_ALIGNMENT_FOR_EXPRESSIONS_IN_SWITCH_CASE_WITH_ARROW	FORMATTER / Option for alignment of expressions in switch case with arrow - option id: "org.eclipse.jdt.core.formatter.alignment_for_expressions_in_switch_case_ - possible values: values returned by `createAlignmentValue(boolean, int, int)` call - default: createAlignmentValue(false, WRAP_COMPACT, INDENT_DEFAULT)
static final String	FORMATTER_ALIGNMENT_FOR_EXPRESSIONS_IN_SWITCH_CASE_WITH_COLON	FORMATTER / Option for alignment of expressions in switch case with colon - option id: "org.eclipse.jdt.core.formatter.alignment_for_expressions_in_switch_case_ - possible values: values returned by `createAlignmentValue(boolean, int, int)` call - default: createAlignmentValue(false, WRAP_COMPACT, INDENT_DEFAULT)

static final String	FORMATTER_ALIGNMENT_FOR_LOGICAL_OPERATOR	FORMATTER / Option for alignment of expressions with logical operators (&&, ||) - option id: "org.eclipse.jdt.core.formatter.alignment_for_logical_operator" - possible values: values returned by `createAlignmentValue(boolean, int, int)` call - default: createAlignmentValue(false, WRAP_COMPACT, INDENT_DEFAULT)
static final String	FORMATTER_ALIGNMENT_FOR_METHOD_DECLARATION	FORMATTER / Option for alignment of method declaration - option id: "org.eclipse.jdt.core.formatter.alignment_for_method_declaration" - possible values: values returned by `createAlignmentValue(boolean, int, int)` call - default: createAlignmentValue(false, WRAP_NO_SPLIT, INDENT_DEFAULT)
static final String	FORMATTER_ALIGNMENT_FOR_MODULE_STATEMENTS	FORMATTER / Option for alignment of module statements - option id: "org.eclipse.jdt.core.formatter.alignment_for_module_statements" - possible values: values returned by `createAlignmentValue(boolean, int, int)` call - default: createAlignmentValue(false, WRAP_COMPACT, INDENT_DEFAULT)
static final String	FORMATTER_ALIGNMENT_FOR_MULTIPLE_FIELDS	FORMATTER / Option for alignment of multiple fields - option id: "org.eclipse.jdt.core.formatter.alignment_for_multiple_fields" - possible values: values returned by `createAlignmentValue(boolean, int, int)` call - default: createAlignmentValue(false, WRAP_COMPACT, INDENT_DEFAULT)
static final String	FORMATTER_ALIGNMENT_FOR_MULTIPLICATIVE_OPERATOR	FORMATTER / Option for alignment of expressions with multiplicative operators (*, /, %) - option id: "org.eclipse.jdt.core.formatter.alignment_for_multiplicative_operator" - possible values: values returned by `createAlignmentValue(boolean, int, int)` call - default: createAlignmentValue(false, WRAP_COMPACT, INDENT_DEFAULT)
static final String	FORMATTER_ALIGNMENT_FOR_PARAMETERIZED_TYPE_REFERENCES	FORMATTER / Option for alignment of type arguments in parameterized type references - option id: "org.eclipse.jdt.core.formatter.alignment_for_parameterized_type_referenc - possible values: values returned by `createAlignmentValue(boolean, int, int)` call - default: createAlignmentValue(false, WRAP_NO_SPLIT, INDENT_DEFAULT)
static final String	FORMATTER_ALIGNMENT_FOR_PARAMETERS_IN_CONSTRUCTOR_DECLARATION	FORMATTER / Option for alignment of parameters in constructor declaration - option id: "org.eclipse.jdt.core.formatter.alignment_for_parameters_in_constructor_d - possible values: values returned by `createAlignmentValue(boolean, int, int)` call - default: createAlignmentValue(false, WRAP_COMPACT, INDENT_DEFAULT)
static final String	FORMATTER_ALIGNMENT_FOR_PARAMETERS_IN_METHOD_DECLARATION	FORMATTER / Option for alignment of parameters in method declaration - option id: "org.eclipse.jdt.core.formatter.alignment_for_parameters_in_method_decla - possible values: values returned by `createAlignmentValue(boolean, int, int)` call - default: createAlignmentValue(false, WRAP_COMPACT, INDENT_DEFAULT)
static final String	FORMATTER_ALIGNMENT_FOR_PERMITTED_TYPES_IN_TYPE_DECLARATION	FORMATTER / Option for alignment of permitted types in type declaration - option id: "org.eclipse.jdt.core.formatter.alignment_for_permitted_types_in_type_dec - possible values: values returned by `createAlignmentValue(boolean, int, int)` call - default: createAlignmentValue(false, WRAP_COMPACT, INDENT_DEFAULT)
static final String	FORMATTER_ALIGNMENT_FOR_RECORD_COMPONENTS	FORMATTER / Option for alignment of components in record declaration - option id: "org.eclipse.jdt.core.formatter.alignment_for_record_components" - possible values: values returned by `createAlignmentValue(boolean, int, int)` call - default: createAlignmentValue(false, WRAP_COMPACT, INDENT_DEFAULT)
static final String	FORMATTER_ALIGNMENT_FOR_RELATIONAL_OPERATOR	FORMATTER / Option for alignment of expressions with relational operators (<, >, <=, >=, ==, !=) - option id: "org.eclipse.jdt.core.formatter.alignment_for_relational_operator" - possible values: values returned by `createAlignmentValue(boolean, int, int)` call - default: createAlignmentValue(false, WRAP_COMPACT, INDENT_DEFAULT)
static final String	FORMATTER_ALIGNMENT_FOR_RESOURCES_IN_TRY	FORMATTER / Option for alignment of resources in a try with resources statement - option id: "org.eclipse.jdt.core.formatter.alignment_for_resources_in_try" - possible values: values returned by `createAlignmentValue(boolean, int, int)` call - default: createAlignmentValue(false, WRAP_NEXT_PER_LINE, INDENT_DEFAULT)

static final String	FORMATTER_ALIGNMENT_FOR_SELECTOR_IN_METHOD_INVOCATION	FORMATTER / Option for alignment of selector in method invocation - option id: "org.eclipse.jdt.core.formatter.alignment_for_selector_in_method_invocati - possible values: values returned by `createAlignmentValue(boolean, int, int)` call - default: createAlignmentValue(false, WRAP_COMPACT, INDENT_DEFAULT)
static final String	FORMATTER_ALIGNMENT_FOR_SHIFT_OPERATOR	FORMATTER / Option for alignment of expressions with shift operators (<<, >>, >>>) - option id: "org.eclipse.jdt.core.formatter.alignment_for_shift_operator" - possible values: values returned by `createAlignmentValue(boolean, int, int)` call - default: createAlignmentValue(false, WRAP_COMPACT, INDENT_DEFAULT)
static final String	FORMATTER_ALIGNMENT_FOR_STRING_CONCATENATION	FORMATTER / Option for alignment of string concatenation expressions - option id: "org.eclipse.jdt.core.formatter.alignment_for_string_concatenation" - possible values: values returned by `createAlignmentValue(boolean, int, int)` call - default: createAlignmentValue(false, WRAP_COMPACT, INDENT_DEFAULT)
static final String	FORMATTER_ALIGNMENT_FOR_SUPERCLASS_IN_TYPE_DECLARATION	FORMATTER / Option for alignment of superclass in type declaration - option id: "org.eclipse.jdt.core.formatter.alignment_for_superclass_in_type_declarati - possible values: values returned by `createAlignmentValue(boolean, int, int)` call - default: createAlignmentValue(false, WRAP_NEXT_SHIFTED, INDENT_DEFAULT)
static final String	FORMATTER_ALIGNMENT_FOR_SUPERINTERFACES_IN_ENUM_DECLARATION	FORMATTER / Option for alignment of superinterfaces in enum declaration - option id: "org.eclipse.jdt.core.formatter.alignment_for_superinterfaces_in_enum_de - possible values: values returned by `createAlignmentValue(boolean, int, int)` call - default: createAlignmentValue(false, WRAP_COMPACT, INDENT_DEFAULT)
static final String	FORMATTER_ALIGNMENT_FOR_SUPERINTERFACES_IN_RECORD_DECLARATION	FORMATTER / Option for alignment of superinterfaces in record declaration - option id: "org.eclipse.jdt.core.formatter.alignment_for_superinterfaces_in_record_de - possible values: values returned by `createAlignmentValue(boolean, int, int)` call - default: createAlignmentValue(false, WRAP_COMPACT, INDENT_DEFAULT)
static final String	FORMATTER_ALIGNMENT_FOR_SUPERINTERFACES_IN_TYPE_DECLARATION	FORMATTER / Option for alignment of superinterfaces in type declaration - option id: "org.eclipse.jdt.core.formatter.alignment_for_superinterfaces_in_type_decl - possible values: values returned by `createAlignmentValue(boolean, int, int)` call - default: createAlignmentValue(false, WRAP_COMPACT, INDENT_DEFAULT)
static final String	FORMATTER_ALIGNMENT_FOR_SWITCH_CASE_WITH_ARROW	FORMATTER / Option for alignment of arrow in switch case (->) - option id: "org.eclipse.jdt.core.formatter.alignment_for_switch_case_with_arrow" - possible values: values returned by `createAlignmentValue(boolean, int, int)` call - default: createAlignmentValue(false, WRAP_COMPACT, INDENT_BY_ONE)
static final String	FORMATTER_ALIGNMENT_FOR_THROWS_CLAUSE_IN_CONSTRUCTOR_DECLARATI	FORMATTER / Option for alignment of throws clause in constructor declaration - option id: "org.eclipse.jdt.core.formatter.alignment_for_throws_clause_in_constructo - possible values: values returned by `createAlignmentValue(boolean, int, int)` call - default: createAlignmentValue(false, WRAP_COMPACT, INDENT_DEFAULT)
static final String	FORMATTER_ALIGNMENT_FOR_THROWS_CLAUSE_IN_METHOD_DECLARATION	FORMATTER / Option for alignment of throws clause in method declaration - option id: "org.eclipse.jdt.core.formatter.alignment_for_throws_clause_in_method_d - possible values: values returned by `createAlignmentValue(boolean, int, int)` call - default: createAlignmentValue(false, WRAP_COMPACT, INDENT_DEFAULT)
static final String	FORMATTER_ALIGNMENT_FOR_TYPE_ANNOTATIONS	FORMATTER / Option for alignment of type annotations - option id: "org.eclipse.jdt.core.formatter.alignment_for_type_annotations" - possible values: values returned by `createAlignmentValue(boolean, int)` call - default: createAlignmentValue(false, WRAP_NO_SPLIT)
static final String	FORMATTER_ALIGNMENT_FOR_TYPE_ARGUMENTS	FORMATTER / Option for alignment of type arguments in method invocations and references - option id: "org.eclipse.jdt.core.formatter.alignment_for_type_arguments" - possible values: values returned by `createAlignmentValue(boolean, int, int)` call - default: createAlignmentValue(false, WRAP_NO_SPLIT, INDENT_DEFAULT)

static final String	FORMATTER_ALIGNMENT_FOR_TYPE_PARAMETERS	FORMATTER / Option for alignment of type parameters in method and type declarations - option id: "org.eclipse.jdt.core.formatter.alignment_for_type_parameters" - possible values: values returned by `createAlignmentValue(boolean, int, int)` call - default: createAlignmentValue(false, WRAP_NO_SPLIT, INDENT_DEFAULT)
static final String	FORMATTER_ALIGNMENT_FOR_UNION_TYPE_IN_MULTICATCH	FORMATTER / Option for alignment of exceptions declared in a Union Type in the argument of a multicatch statement - option id: "org.eclipse.jdt.core.formatter.alignment_for_union_type_in_multicatch" - possible values: values returned by `createAlignmentValue(boolean, int, int)` call - default: createAlignmentValue(false, WRAP_COMPACT, INDENT_DEFAULT)
static final String	FORMATTER_BLANK_LINES_AFTER_CODE_BLOCK	FORMATTER / Option to add or remove blank lines after a statement containing a code block - option id: "org.eclipse.jdt.core.formatter.number_of_blank_lines_after_code_block" - possible values: "<n>", where n is an integer.
static final String	FORMATTER_BLANK_LINES_AFTER_IMPORTS	FORMATTER / Option to add or remove blank lines after the imports declaration - option id: "org.eclipse.jdt.core.formatter.blank_lines_after_imports" - possible values: "<n>", where n is an integer.
static final String	FORMATTER_BLANK_LINES_AFTER_LAST_CLASS_BODY_DECLARATION	FORMATTER / Option to add or remove blank lines after the last class body declaration - option id: "org.eclipse.jdt.core.formatter.blank_lines_after_last_class_body_declaratio - possible values: "<n>", where n is an integer.
static final String	FORMATTER_BLANK_LINES_AFTER_PACKAGE	FORMATTER / Option to add or remove blank lines after the package declaration - option id: "org.eclipse.jdt.core.formatter.blank_lines_after_package" - possible values: "<n>", where n is an integer.
static final String	FORMATTER_BLANK_LINES_AT_BEGINNING_OF_CODE_BLOCK	FORMATTER / Option to add or remove blank lines at the beginning of the code block - option id: "org.eclipse.jdt.core.formatter.number_of_blank_lines_at_beginning_of_co - possible values: "<n>", where n is an integer.
static final String	FORMATTER_BLANK_LINES_AT_BEGINNING_OF_METHOD_BODY	FORMATTER / Option to add or remove blank lines at the beginning of the method body - option id: "org.eclipse.jdt.core.formatter.number_of_blank_lines_at_beginning_of_m - possible values: "<n>", where n is an integer.
static final String	FORMATTER_BLANK_LINES_AT_END_OF_CODE_BLOCK	FORMATTER / Option to add or remove blank lines at the end of the code block - option id: "org.eclipse.jdt.core.formatter.number_of_blank_lines_at_end_of_code_bl - possible values: "<n>", where n is an integer.
static final String	FORMATTER_BLANK_LINES_AT_END_OF_METHOD_BODY	FORMATTER / Option to add or remove blank lines at the end of the method body - option id: "org.eclipse.jdt.core.formatter.number_of_blank_lines_at_end_of_method_ - possible values: "<n>", where n is an integer.
static final String	FORMATTER_BLANK_LINES_BEFORE_ABSTRACT_METHOD	FORMATTER / Option to add or remove blank lines before an abstract method declaration - option id: "org.eclipse.jdt.core.formatter.blank_lines_before_abstract_method" - possible values: "<n>", where n is an integer.
static final String	FORMATTER_BLANK_LINES_BEFORE_CODE_BLOCK	FORMATTER / Option to add or remove blank lines before a statement containing a code block - option id: "org.eclipse.jdt.core.formatter.number_of_blank_lines_before_code_block" - possible values: "<n>", where n is an integer.
static final String	FORMATTER_BLANK_LINES_BEFORE_FIELD	FORMATTER / Option to add or remove blank lines before a field declaration - option id: "org.eclipse.jdt.core.formatter.blank_lines_before_field" - possible values: "<n>", where n is an integer.
static final String	FORMATTER_BLANK_LINES_BEFORE_FIRST_CLASS_BODY_DECLARATION	FORMATTER / Option to add or remove blank lines before the first class body declaration - option id: "org.eclipse.jdt.core.formatter.blank_lines_before_first_class_body_declara - possible values: "<n>", where n is an integer.
static final String	FORMATTER_BLANK_LINES_BEFORE_IMPORTS	FORMATTER / Option to add or remove blank lines before the imports declaration - option id: "org.eclipse.jdt.core.formatter.blank_lines_before_imports" - possible values: "<n>", where n is an integer.
static final String	FORMATTER_BLANK_LINES_BEFORE_MEMBER_TYPE	FORMATTER / Option to add or remove blank lines before a member type declaration - option id: "org.eclipse.jdt.core.formatter.blank_lines_before_member_type" - possible values: "<n>", where n is an integer.

static final String	FORMATTER_BLANK_LINES_BEFORE_METHOD	FORMATTER / Option to add or remove blank lines before a non-abstract method declaration - option id: "org.eclipse.jdt.core.formatter.blank_lines_before_method" - possible values: "<n>", where n is an integer.
static final String	FORMATTER_BLANK_LINES_BEFORE_NEW_CHUNK	FORMATTER / Option to add or remove blank lines before a new chunk - option id: "org.eclipse.jdt.core.formatter.blank_lines_before_new_chunk" - possible values: "<n>", where n is an integer.
static final String	FORMATTER_BLANK_LINES_BEFORE_PACKAGE	FORMATTER / Option to add or remove blank lines before the package declaration - option id: "org.eclipse.jdt.core.formatter.blank_lines_before_package" - possible values: "<n>", where n is an integer.
static final String	FORMATTER_BLANK_LINES_BETWEEN_IMPORT_GROUPS	FORMATTER / Option to add or remove blank lines between import groups - option id: "org.eclipse.jdt.core.formatter.blank_lines_between_import_groups" - possible values: "<n>", where n is an integer.
static final String	FORMATTER_BLANK_LINES_BETWEEN_STATEMENT_GROUPS_IN_SWITCH	FORMATTER / Option to add or remove blank lines between statement groups in switch - option id: "org.eclipse.jdt.core.formatter.blank_lines_between_statement_groups_in_ - possible values: "<n>", where n is an integer.
static final String	FORMATTER_BLANK_LINES_BETWEEN_TYPE_DECLARATIONS	FORMATTER / Option to add or remove blank lines between type declarations - option id: "org.eclipse.jdt.core.formatter.blank_lines_between_type_declarations" - possible values: "<n>", where n is an integer.
static final String	FORMATTER_BRACE_POSITION_FOR_ANNOTATION_TYPE_DECLARATION	FORMATTER / Option to position the braces of an annotation type declaration - option id: "org.eclipse.jdt.core.formatter.brace_position_for_annotation_type_declara - possible values: { END_OF_LINE, NEXT_LINE, NEXT_LINE_SHIFTED, NEXT_LINE_ON_WRAP } - default: END_OF_LINE
static final String	FORMATTER_BRACE_POSITION_FOR_ANONYMOUS_TYPE_DECLARATION	FORMATTER / Option to position the braces of an anonymous type declaration - option id: "org.eclipse.jdt.core.formatter.brace_position_for_anonymous_type_declara - possible values: { END_OF_LINE, NEXT_LINE, NEXT_LINE_SHIFTED, NEXT_LINE_ON_WRAP } - default: END_OF_LINE
static final String	FORMATTER_BRACE_POSITION_FOR_ARRAY_INITIALIZER	FORMATTER / Option to position the braces of an array initializer - option id: "org.eclipse.jdt.core.formatter.brace_position_for_array_initializer" - possible values: { END_OF_LINE, NEXT_LINE, NEXT_LINE_SHIFTED, NEXT_LINE_ON_WRAP } - default: END_OF_LINE
static final String	FORMATTER_BRACE_POSITION_FOR_BLOCK	FORMATTER / Option to position the braces of a block - option id: "org.eclipse.jdt.core.formatter.brace_position_for_block" - possible values: { END_OF_LINE, NEXT_LINE, NEXT_LINE_SHIFTED, NEXT_LINE_ON_WRAP } - default: END_OF_LINE
static final String	FORMATTER_BRACE_POSITION_FOR_BLOCK_IN_CASE	FORMATTER / Option to position the braces of a block in a switch statement/expression when the block is the first statement following a case with colon - option id: "org.eclipse.jdt.core.formatter.brace_position_for_block_in_case" - possible values: { END_OF_LINE, NEXT_LINE, NEXT_LINE_SHIFTED, NEXT_LINE_ON_WRAP } - default: END_OF_LINE
static final String	FORMATTER_BRACE_POSITION_FOR_BLOCK_IN_CASE_AFTER_ARROW	FORMATTER / Option to position the braces of a block in a switch statement/expression when the block is the first statement following a case with arrow - option id: "org.eclipse.jdt.core.formatter.brace_position_for_block_in_case_after_arr - possible values: { END_OF_LINE, NEXT_LINE, NEXT_LINE_SHIFTED, NEXT_LINE_ON_WRAP } - default: END_OF_LINE
static final String	FORMATTER_BRACE_POSITION_FOR_CONSTRUCTOR_DECLARATION	FORMATTER / Option to position the braces of a constructor declaration - option id: "org.eclipse.jdt.core.formatter.brace_position_for_constructor_declaration" - possible values: { END_OF_LINE, NEXT_LINE, NEXT_LINE_SHIFTED, NEXT_LINE_ON_WRAP } - default: END_OF_LINE

static final String	FORMATTER_BRACE_POSITION_FOR_ENUM_CONSTANT	FORMATTER / Option to position the braces of an enum constant - option id: "org.eclipse.jdt.core.formatter.brace_position_for_enum_constant" - possible values: { END_OF_LINE, NEXT_LINE, NEXT_LINE_SHIFTED, NEXT_LINE_ON_WRAP } - default: END_OF_LINE
static final String	FORMATTER_BRACE_POSITION_FOR_ENUM_DECLARATION	FORMATTER / Option to position the braces of an enum declaration - option id: "org.eclipse.jdt.core.formatter.brace_position_for_enum_declaration" - possible values: { END_OF_LINE, NEXT_LINE, NEXT_LINE_SHIFTED, NEXT_LINE_ON_WRAP } - default: END_OF_LINE
static final String	FORMATTER_BRACE_POSITION_FOR_LAMBDA_BODY	FORMATTER / Option to position the braces of a lambda block - option id: "org.eclipse.jdt.core.formatter.brace_position_for_lambda_body" - possible values: { END_OF_LINE, NEXT_LINE, NEXT_LINE_SHIFTED, NEXT_LINE_ON_WRAP } - default: END_OF_LINE
static final String	FORMATTER_BRACE_POSITION_FOR_METHOD_DECLARATION	FORMATTER / Option to position the braces of a method declaration - option id: "org.eclipse.jdt.core.formatter.brace_position_for_method_declaration" - possible values: { END_OF_LINE, NEXT_LINE, NEXT_LINE_SHIFTED, NEXT_LINE_ON_WRAP } - default: END_OF_LINE
static final String	FORMATTER_BRACE_POSITION_FOR_RECORD_CONSTRUCTOR	FORMATTER / Option to position the braces of a record constructor - option id: "org.eclipse.jdt.core.formatter.brace_position_for_record_constructor" - possible values: { END_OF_LINE, NEXT_LINE, NEXT_LINE_SHIFTED, NEXT_LINE_ON_WRAP } - default: END_OF_LINE
static final String	FORMATTER_BRACE_POSITION_FOR_RECORD_DECLARATION	FORMATTER / Option to position the braces of a record declaration - option id: "org.eclipse.jdt.core.formatter.brace_position_for_record_declaration" - possible values: { END_OF_LINE, NEXT_LINE, NEXT_LINE_SHIFTED, NEXT_LINE_ON_WRAP } - default: END_OF_LINE
static final String	FORMATTER_BRACE_POSITION_FOR_SWITCH	FORMATTER / Option to position the braces of a switch statement - option id: "org.eclipse.jdt.core.formatter.brace_position_for_switch" - possible values: { END_OF_LINE, NEXT_LINE, NEXT_LINE_SHIFTED, NEXT_LINE_ON_WRAP } - default: END_OF_LINE
static final String	FORMATTER_BRACE_POSITION_FOR_TYPE_DECLARATION	FORMATTER / Option to position the braces of a type declaration - option id: "org.eclipse.jdt.core.formatter.brace_position_for_type_declaration" - possible values: { END_OF_LINE, NEXT_LINE, NEXT_LINE_SHIFTED, NEXT_LINE_ON_WRAP } - default: END_OF_LINE
static final String	FORMATTER_COMMENT_ALIGN_TAGS_DESCREIPTIONS_GROUPED	FORMATTER / Option to control whether descriptions and names in Javadoc root tags, should be aligned and grouped by tag type
static final String	FORMATTER_COMMENT_ALIGN_TAGS_NAMES_DESCRIPTIONS	FORMATTER / Option to control whether names and descriptions in Javadoc root tags should be aligned
static final String	FORMATTER_COMMENT_CLEAR_BLANK_LINES	**Deprecated.** Use FORMATTER_COMMENT_CLEAR_BLANK_LINES_IN_BLOCK_COMMENT and FORMATTER_COMMENT_CLEAR_BLANK_LINES_IN_JAVADOC_COMMENT
static final String	FORMATTER_COMMENT_CLEAR_BLANK_LINES_IN_BLOCK_COMMENT	FORMATTER / Option to control whether blank lines are cleared inside block comments - option id: "org.eclipse.jdt.core.formatter.comment.clear_blank_lines_in_block_comm - possible values: { TRUE, FALSE } - default: FALSE
static final String	FORMATTER_COMMENT_CLEAR_BLANK_LINES_IN_JAVADOC_COMMENT	FORMATTER / Option to control whether blank lines are cleared inside javadoc comments - option id: "org.eclipse.jdt.core.formatter.comment.clear_blank_lines_in_javadoc_com - possible values: { TRUE, FALSE } - default: FALSE
static final String	FORMATTER_COMMENT_COUNT_LINE_LENGTH_FROM_STARTING_POSITION	FORMATTER / Option to control whether comments' line length will be counted from their starting position - option id: "org.eclipse.jdt.core.formatter.comment.count_line_length_from_starting_ - possible values: { TRUE, FALSE } - default: TRUE
static final String	FORMATTER_COMMENT_FORMAT	**Deprecated.** Use multiple settings for each kind of comments.

static final String	FORMATTER_COMMENT_FORMAT_BLOCK_COMMENT	FORMATTER / Option to control whether multiple lines comments are formatted - option id: "org.eclipse.jdt.core.formatter.comment.format_block_comments" - possible values: { TRUE, FALSE } - default: TRUE
static final String	FORMATTER_COMMENT_FORMAT_HEADER	FORMATTER / Option to control whether the header comment of a Java source file is formatted - option id: "org.eclipse.jdt.core.formatter.comment.format_header" - possible values: { TRUE, FALSE } - default: FALSE
static final String	FORMATTER_COMMENT_FORMAT_HTML	FORMATTER / Option to control whether HTML tags are formatted
static final String	FORMATTER_COMMENT_FORMAT_JAVADOC_COMMENT	FORMATTER / Option to control whether javadoc comments are formatted - option id: "org.eclipse.jdt.core.formatter.comment.format_javadoc_comments" - possible values: { TRUE, FALSE } - default: TRUE
static final String	FORMATTER_COMMENT_FORMAT_LINE_COMMENT	FORMATTER / Option to control whether single line comments are formatted - option id: "org.eclipse.jdt.core.formatter.comment.format_line_comments" - possible values: { TRUE, FALSE } - default: TRUE
static final String	FORMATTER_COMMENT_FORMAT_LINE_COMMENT_STARTING_ON_FIRST_COLUMN	FORMATTER / Option to format line comments that start on the first column - option id: "org.eclipse.jdt.core.formatter.format_line_comment_starting_on_first_col - possible values: { TRUE, FALSE } - default: TRUE
static final String	FORMATTER_COMMENT_FORMAT_MARKDOWN_COMMENT	FORMATTER / Option to control whether markdown comments are formatted - option id: "org.eclipse.jdt.core.formatter.comment.format_markdown_comments" - possible values: { TRUE, FALSE } - default: TRUE
static final String	FORMATTER_COMMENT_FORMAT_SOURCE	FORMATTER / Option to control whether code snippets are formatted in comments - option id: "org.eclipse.jdt.core.formatter.comment.format_source_code" - possible values: { TRUE, FALSE } - default: TRUE
static final String	FORMATTER_COMMENT_INDENT_PARAMETER_DESCRIPTION	FORMATTER / Option to control whether description of Javadoc parameters are indented - option id: "org.eclipse.jdt.core.formatter.comment.indent_parameter_description" - possible values: { TRUE, FALSE } - default: TRUE
static final String	FORMATTER_COMMENT_INDENT_ROOT_TAGS	FORMATTER / Option to control whether Javadoc root tags are indented
static final String	FORMATTER_COMMENT_INDENT_TAG_DESCRIPTION	FORMATTER / Option to control whether Javadoc tag descriptions are indented when wrapped, excluding tags controlled by #FORMATTER_COMMENT_INDENT_PARAMETER_DESCRIPTION - option id: "org.eclipse.jdt.core.formatter.comment.indent_return_description" - possible values: { TRUE, FALSE } - default: TRUE
static final String	FORMATTER_COMMENT_INSERT_EMPTY_LINE_BEFORE_ROOT_TAGS	FORMATTER / Option to insert an empty line before the Javadoc root tag block - option id: "org.eclipse.jdt.core.formatter.comment.insert_new_line_before_root_tags" - possible values: { INSERT, DO_NOT_INSERT } - default: INSERT
static final String	FORMATTER_COMMENT_INSERT_EMPTY_LINE_BETWEEN_DIFFERENT_TAGS	FORMATTER / Option to insert an empty line between Javadoc tags of different type - option id: "org.eclipse.jdt.core.formatter.comment.insert_new_line_between_different - possible values: { INSERT, DO_NOT_INSERT } - default: INSERT
static final String	FORMATTER_COMMENT_INSERT_NEW_LINE_FOR_PARAMETER	FORMATTER / Option to insert a new line after Javadoc root tag parameters - option id: "org.eclipse.jdt.core.formatter.comment.insert_new_line_for_parameter" - possible values: { INSERT, DO_NOT_INSERT } - default: INSERT
static final String	FORMATTER_COMMENT_JAVADOC_DO_NOT_SEPARATE_BLOCK_TAGS	FORMATTER / Option to control whether paragraph tags in javadoc comments are put with the content or on their own line - option id: "org.eclipse.jdt.core.formatter.comment.javadoc_paragraphs_tags_with_cor - possible values: { TRUE, FALSE } - default: FALSE
static final String	FORMATTER_COMMENT_LINE_LENGTH	FORMATTER / Option to specify the line length for comments
static final String	FORMATTER_COMMENT_NEW_LINES_AT_BLOCK_BOUNDARIES	FORMATTER / Option to control whether block comments will have new lines at boundaries - option id: "org.eclipse.jdt.core.formatter.comment.new_lines_at_block_boundaries" - possible values: { TRUE, FALSE } - default: TRUE
static final String	FORMATTER_COMMENT_NEW_LINES_AT_JAVADOC_BOUNDARIES	FORMATTER / Option to control whether javadoc comments will have new lines at boundaries - option id: "org.eclipse.jdt.core.formatter.comment.new_lines_at_javadoc_boundaries" - possible values: { TRUE, FALSE } - default: TRUE

static final String	FORMATTER_COMMENT_PRESERVE_WHITE_SPACE_BETWEEN_CODE_AND_LINE_C	FORMATTER / Option to control whether the white space between code and line comments should be preserved or replaced with a single space - option id: "org.eclipse.jdt.core.formatter.comment.preserve_white_space_between_co - possible values: { TRUE, FALSE } - default: FALSE
static final String	FORMATTER_COMPACT_ELSE_IF	FORMATTER / Option to compact else/if - option id: "org.eclipse.jdt.core.formatter.compact_else_if" - possible values: { TRUE, FALSE } - default: TRUE
static final String	FORMATTER_CONTINUATION_INDENTATION	FORMATTER / Option to set the continuation indentation - option id: "org.eclipse.jdt.core.formatter.continuation_indentation" - possible values: "<n>", where n is zero or a positive integer - default: "2"
static final String	FORMATTER_CONTINUATION_INDENTATION_FOR_ARRAY_INITIALIZER	FORMATTER / Option to set the continuation indentation inside array initializer - option id: "org.eclipse.jdt.core.formatter.continuation_indentation_for_array_initializ - possible values: "<n>", where n is zero or a positive integer - default: "2"
static final String	FORMATTER_DISABLING_TAG	FORMATTER / Option to define the tag to put in a comment to disable the formatting
static final String	FORMATTER_ENABLING_TAG	FORMATTER / Option to define the tag to put in a comment to re-enable the formatting after it has been disabled (see FORMATTER_DISABLING_TAG) - option id: "org.eclipse.jdt.core.formatter.enabling_tag" - possible values: String, with constraints mentioned below - default: "@formatter:on"
static final String	FORMATTER_INDENT_BODY_DECLARATIONS_COMPARE_TO_ANNOTATION_DECLA	FORMATTER / Option to indent body declarations compare to its enclosing annotation declaration header - option id: "org.eclipse.jdt.core.formatter.indent_body_declarations_compare_to_annc - possible values: { TRUE, FALSE } - default: TRUE
static final String	FORMATTER_INDENT_BODY_DECLARATIONS_COMPARE_TO_ENUM_CONSTANT_HE	FORMATTER / Option to indent body declarations compare to its enclosing enum constant header - option id: "org.eclipse.jdt.core.formatter.indent_body_declarations_compare_to_enur - possible values: { TRUE, FALSE } - default: TRUE
static final String	FORMATTER_INDENT_BODY_DECLARATIONS_COMPARE_TO_ENUM_DECLARATION	FORMATTER / Option to indent body declarations compare to its enclosing enum declaration header - option id: "org.eclipse.jdt.core.formatter.indent_body_declarations_compare_to_enur - possible values: { TRUE, FALSE } - default: TRUE
static final String	FORMATTER_INDENT_BODY_DECLARATIONS_COMPARE_TO_RECORD_HEADER	FORMATTER / Option to indent body declarations compare to its enclosing record header - option id: "org.eclipse.jdt.core.formatter.indent_body_declarations_compare_to_reco - possible values: { TRUE, FALSE } - default: TRUE
static final String	FORMATTER_INDENT_BODY_DECLARATIONS_COMPARE_TO_TYPE_HEADER	FORMATTER / Option to indent body declarations compare to its enclosing type header - option id: "org.eclipse.jdt.core.formatter.indent_body_declarations_compare_to_type - possible values: { TRUE, FALSE } - default: TRUE
static final String	FORMATTER_INDENT_BREAKS_COMPARE_TO_CASES	FORMATTER / Option to indent breaks compare to cases - option id: "org.eclipse.jdt.core.formatter.indent_breaks_compare_to_cases" - possible values: { TRUE, FALSE } - default: TRUE
static final String	FORMATTER_INDENT_EMPTY_LINES	FORMATTER / Option to indent empty lines - option id: "org.eclipse.jdt.core.formatter.indent_empty_lines" - possible values: { TRUE, FALSE } - default: FALSE
static final String	FORMATTER_INDENT_STATEMENTS_COMPARE_TO_BLOCK	FORMATTER / Option to indent statements inside a block - option id: "org.eclipse.jdt.core.formatter.indent_statements_compare_to_block" - possible values: { TRUE, FALSE } - default: TRUE
static final String	FORMATTER_INDENT_STATEMENTS_COMPARE_TO_BODY	FORMATTER / Option to indent statements inside the body of a method or a constructor - option id: "org.eclipse.jdt.core.formatter.indent_statements_compare_to_body" - possible values: { TRUE, FALSE } - default: TRUE
static final String	FORMATTER_INDENT_SWITCHSTATEMENTS_COMPARE_TO_CASES	FORMATTER / Option to indent switch statements compare to cases - option id: "org.eclipse.jdt.core.formatter.indent_switchstatements_compare_to_cases" - possible values: { TRUE, FALSE } - default: TRUE
static final String	FORMATTER_INDENT_SWITCHSTATEMENTS_COMPARE_TO_SWITCH	FORMATTER / Option to indent switch statements compare to switch - option id: "org.eclipse.jdt.core.formatter.indent_switchstatements_compare_to_switch - possible values: { TRUE, FALSE } - default: TRUE
static final String	FORMATTER_INDENTATION_SIZE	FORMATTER / Option to specify the equivalent number of spaces that represents one indentation - option id: "org.eclipse.jdt.core.formatter.indentation.size" - possible values: "<n>", where n is zero or a positive integer - default: "4"

static final String	FORMATTER_INSERT_NEW_LINE_AFTER_ANNOTATION	**Deprecated.** All new options must be enabled to activate old strategy FORMATTER_INSERT_NEW_LINE_AFTER_ANNOTATION_ON_MEMBER FORMATTER_INSERT_NEW_LINE_AFTER_ANNOTATION_ON_LOCAL_VARIABLE FORMATTER_INSERT_NEW_LINE_AFTER_ANNOTATION_ON_PARAMETER
static final String	FORMATTER_INSERT_NEW_LINE_AFTER_ANNOTATION_ON_ENUM_CONSTANT	FORMATTER / Option to insert a new line after an annotation on an enum constant declaration - option id: "org.eclipse.jdt.core.formatter.insert_new_line_after_annotation_on_enum - possible values: { INSERT, DO_NOT_INSERT } - default: INSERT
static final String	FORMATTER_INSERT_NEW_LINE_AFTER_ANNOTATION_ON_FIELD	FORMATTER / Option to insert a new line after an annotation on a field declaration - option id: "org.eclipse.jdt.core.formatter.insert_new_line_after_annotation_on_field" - possible values: { INSERT, DO_NOT_INSERT } - default: INSERT
static final String	FORMATTER_INSERT_NEW_LINE_AFTER_ANNOTATION_ON_LOCAL_VARIABLE	FORMATTER / Option to insert a new line after an annotation on a local variable - option id: "org.eclipse.jdt.core.formatter.insert_new_line_after_annotation_on_local_ - possible values: { INSERT, DO_NOT_INSERT } - default: INSERT
static final String	FORMATTER_INSERT_NEW_LINE_AFTER_ANNOTATION_ON_MEMBER	**Deprecated.** All new options must be enabled to activate old strategy FORMATTER_INSERT_NEW_LINE_AFTER_ANNOTATION_ON_FIELD FORMATTER_INSERT_NEW_LINE_AFTER_ANNOTATION_ON_METHOD FORMATTER_INSERT_NEW_LINE_AFTER_ANNOTATION_ON_PACKAGE FORMATTER_INSERT_NEW_LINE_AFTER_ANNOTATION_ON_TYPE
static final String	FORMATTER_INSERT_NEW_LINE_AFTER_ANNOTATION_ON_METHOD	FORMATTER / Option to insert a new line after an annotation on a method declaration - option id: "org.eclipse.jdt.core.formatter.insert_new_line_after_annotation_on_metho - possible values: { INSERT, DO_NOT_INSERT } - default: INSERT
static final String	FORMATTER_INSERT_NEW_LINE_AFTER_ANNOTATION_ON_PACKAGE	FORMATTER / Option to insert a new line after an annotation on a package declaration - option id: "org.eclipse.jdt.core.formatter.insert_new_line_after_annotation_on_packa - possible values: { INSERT, DO_NOT_INSERT } - default: INSERT
static final String	FORMATTER_INSERT_NEW_LINE_AFTER_ANNOTATION_ON_PARAMETER	FORMATTER / Option to insert a new line after an annotation on a parameter - option id: "org.eclipse.jdt.core.formatter.insert_new_line_after_annotation_on_paran - possible values: { INSERT, DO_NOT_INSERT } - default: DO_NOT_INSERT
static final String	FORMATTER_INSERT_NEW_LINE_AFTER_ANNOTATION_ON_TYPE	FORMATTER / Option to insert a new line after an annotation on a type declaration - option id: "org.eclipse.jdt.core.formatter.insert_new_line_after_annotation_on_type" - possible values: { INSERT, DO_NOT_INSERT } - default: INSERT
static final String	FORMATTER_INSERT_NEW_LINE_AFTER_LABEL	FORMATTER / Option to insert a new line after a label - option id: "org.eclipse.jdt.core.formatter.insert_new_line_after_label" - possible values: { INSERT, DO_NOT_INSERT } - default: DO_NOT_INSERT
static final String	FORMATTER_INSERT_NEW_LINE_AFTER_OPENING_BRACE_IN_ARRAY_INITIAL	FORMATTER / Option to insert a new line after the opening brace in an array initializer - option id: "org.eclipse.jdt.core.formatter.insert_new_line_after_opening_brace_in_ar - possible values: { INSERT, DO_NOT_INSERT } - default: DO_NOT_INSERT
static final String	FORMATTER_INSERT_NEW_LINE_AFTER_TYPE_ANNOTATION	FORMATTER / Option to insert a new line after a type annotation - option id: "org.eclipse.jdt.core.formatter.insert_new_line_after_type_annotation" - possible values: { INSERT, DO_NOT_INSERT } - default: DO_NOT_INSERT
static final String	FORMATTER_INSERT_NEW_LINE_AT_END_OF_FILE_IF_MISSING	FORMATTER / Option to insert a new line at the end of the current file if missing - option id: "org.eclipse.jdt.core.formatter.insert_new_line_at_end_of_file_if_missing" - possible values: { INSERT, DO_NOT_INSERT } - default: DO_NOT_INSERT
static final String	FORMATTER_INSERT_NEW_LINE_BEFORE_CATCH_IN_TRY_STATEMENT	FORMATTER / Option to insert a new line before the catch keyword in try statement - option id: "org.eclipse.jdt.core.formatter.insert_new_line_before_catch_in_try_staten - possible values: { INSERT, DO_NOT_INSERT } - default: DO_NOT_INSERT
static final String	FORMATTER_INSERT_NEW_LINE_BEFORE_CLOSING_BRACE_IN_ARRAY_INITIA	FORMATTER / Option to insert a new line before the closing brace in an array initializer - option id: "org.eclipse.jdt.core.formatter.insert_new_line_before_closing_brace_in_a - possible values: { INSERT, DO_NOT_INSERT } - default: DO_NOT_INSERT

static final String	FORMATTER_INSERT_NEW_LINE_BEFORE_ELSE_IN_IF_STATEMENT	FORMATTER / Option to insert a new line before the else keyword in if statement - option id: "org.eclipse.jdt.core.formatter.insert_new_line_before_else_in_if_statemen - possible values: { INSERT, DO_NOT_INSERT } - default: DO_NOT_INSERT
static final String	FORMATTER_INSERT_NEW_LINE_BEFORE_FINALLY_IN_TRY_STATEMENT	FORMATTER / Option to insert a new line before the finally keyword in try statement - option id: "org.eclipse.jdt.core.formatter.insert_new_line_before_finally_in_try_state - possible values: { INSERT, DO_NOT_INSERT } - default: DO_NOT_INSERT
static final String	FORMATTER_INSERT_NEW_LINE_BEFORE_WHILE_IN_DO_STATEMENT	FORMATTER / Option to insert a new line before while in do statement - option id: "org.eclipse.jdt.core.formatter.insert_new_line_before_while_in_do_statem - possible values: { INSERT, DO_NOT_INSERT } - default: DO_NOT_INSERT
static final String	FORMATTER_INSERT_NEW_LINE_IN_EMPTY_ANNOTATION_DECLARATION	**Deprecated.** Use FORMATTER_KEEP_ANNOTATION_DECLARATION_ON_ONE_LINE instead.
static final String	FORMATTER_INSERT_NEW_LINE_IN_EMPTY_ANONYMOUS_TYPE_DECLARATION	**Deprecated.** Use FORMATTER_KEEP_ANONYMOUS_TYPE_DECLARATION_ON_ONE_LINE instead.
static final String	FORMATTER_INSERT_NEW_LINE_IN_EMPTY_BLOCK	**Deprecated.** Use FORMATTER_KEEP_IF_THEN_BODY_BLOCK_ON_ONE_LINE, FORMATTER_KEEP_LOOP_BODY_BLOCK_ON_ONE_LINE, FORMATTER_KEEP_CODE_BLOCK_ON_ONE_LINE, and FORMATTER_KEEP_LAMBDA_BODY_BLOCK_ON_ONE_LINE instead.
static final String	FORMATTER_INSERT_NEW_LINE_IN_EMPTY_ENUM_CONSTANT	**Deprecated.** Use FORMATTER_KEEP_ENUM_CONSTANT_DECLARATION_ON_ONE_LINE instead.
static final String	FORMATTER_INSERT_NEW_LINE_IN_EMPTY_ENUM_DECLARATION	**Deprecated.** Use FORMATTER_KEEP_ENUM_DECLARATION_ON_ONE_LINE instead.
static final String	FORMATTER_INSERT_NEW_LINE_IN_EMPTY_METHOD_BODY	**Deprecated.** Use FORMATTER_KEEP_METHOD_BODY_ON_ONE_LINE instead.
static final String	FORMATTER_INSERT_NEW_LINE_IN_EMPTY_TYPE_DECLARATION	**Deprecated.** Use FORMATTER_KEEP_TYPE_DECLARATION_ON_ONE_LINE instead.
static final String	FORMATTER_INSERT_SPACE_AFTER_ADDITIVE_OPERATOR	FORMATTER / Option to insert a space after an additive operator (+, -) - option id: "org.eclipse.jdt.core.formatter.insert_space_after_additive_operator" - possible values: { INSERT, DO_NOT_INSERT } - default: INSERT
static final String	FORMATTER_INSERT_SPACE_AFTER_AND_IN_TYPE_PARAMETER	FORMATTER / Option to insert a space after and in wilcard - option id: "org.eclipse.jdt.core.formatter.insert_space_after_and_in_type_parameter" - possible values: { INSERT, DO_NOT_INSERT } - default: INSERT
static final String	FORMATTER_INSERT_SPACE_AFTER_ARROW_IN_SWITCH_CASE	FORMATTER / Option to insert a space after arrow in switch case - option id: "org.eclipse.jdt.core.formatter.insert_space_after_arrow_in_switch_case" - possible values: { INSERT, DO_NOT_INSERT } - default: INSERT
static final String	FORMATTER_INSERT_SPACE_AFTER_ARROW_IN_SWITCH_DEFAULT	FORMATTER / Option to insert a space after arrow in switch default - option id: "org.eclipse.jdt.core.formatter.insert_space_after_arrow_in_switch_default - possible values: { INSERT, DO_NOT_INSERT } - default: INSERT
static final String	FORMATTER_INSERT_SPACE_AFTER_ASSIGNMENT_OPERATOR	FORMATTER / Option to insert a space after an assignment operator - option id: "org.eclipse.jdt.core.formatter.insert_space_after_assignment_operator" - possible values: { INSERT, DO_NOT_INSERT } - default: INSERT
static final String	FORMATTER_INSERT_SPACE_AFTER_AT_IN_ANNOTATION	FORMATTER / Option to insert a space after at in annotation - option id: "org.eclipse.jdt.core.formatter.insert_space_after_at_in_annotation" - possible values: { INSERT, DO_NOT_INSERT } - default: INSERT
static final String	FORMATTER_INSERT_SPACE_AFTER_AT_IN_ANNOTATION_TYPE_DECLARATION	FORMATTER / Option to insert a space after at in annotation type declaration - option id: "org.eclipse.jdt.core.formatter.insert_space_after_at_in_annotation_type_d - possible values: { INSERT, DO_NOT_INSERT } - default: DO_NOT_INSERT

static final String	FORMATTER_INSERT_SPACE_AFTER_BINARY_OPERATOR	**Deprecated.** Use the new settings instead: FORMATTER_INSERT_SPACE_AFTER_MULTIPLICATIVE_OPERATOR, FORMATTER_INSERT_SPACE_AFTER_ADDITIVE_OPERATOR, FORMATTER_INSERT_SPACE_AFTER_STRING_CONCATENATION, FORMATTER_INSERT_SPACE_AFTER_SHIFT_OPERATOR, FORMATTER_INSERT_SPACE_AFTER_RELATIONAL_OPERATOR, FORMATTER_INSERT_SPACE_AFTER_BITWISE_OPERATOR, FORMATTER_INSERT_SPACE_AFTER_LOGICAL_OPERATOR
static final String	FORMATTER_INSERT_SPACE_AFTER_BITWISE_OPERATOR	FORMATTER / Option to insert a space after a bitwise operator (&, ^, |) - option id: "org.eclipse.jdt.core.formatter.insert_space_after_bitwise_operator" - possible values: { INSERT, DO_NOT_INSERT } - default: INSERT
static final String	FORMATTER_INSERT_SPACE_AFTER_CLOSING_ANGLE_BRACKET_IN_TYPE_ARG	FORMATTER / Option to insert a space after the closing angle bracket in explicit type arguments on method/constructor invocations - option id: "org.eclipse.jdt.core.formatter.insert_space_after_closing_angle_bracket_in - possible values: { INSERT, DO_NOT_INSERT } - default: DO_NOT_INSERT
static final String	FORMATTER_INSERT_SPACE_AFTER_CLOSING_ANGLE_BRACKET_IN_TYPE_PAR	FORMATTER / Option to insert a space after the closing angle bracket in type parameter declarations - option id: "org.eclipse.jdt.core.formatter.insert_space_after_closing_angle_bracket_in - possible values: { INSERT, DO_NOT_INSERT } - default: INSERT
static final String	FORMATTER_INSERT_SPACE_AFTER_CLOSING_BRACE_IN_BLOCK	FORMATTER / Option to insert a space after the closing brace of a block - option id: "org.eclipse.jdt.core.formatter.insert_space_after_closing_brace_in_block" - possible values: { INSERT, DO_NOT_INSERT } - default: INSERT
static final String	FORMATTER_INSERT_SPACE_AFTER_CLOSING_PAREN_IN_CAST	FORMATTER / Option to insert a space after the closing parenthesis of a cast expression - option id: "org.eclipse.jdt.core.formatter.insert_space_after_closing_paren_in_cast" - possible values: { INSERT, DO_NOT_INSERT } - default: INSERT
static final String	FORMATTER_INSERT_SPACE_AFTER_COLON_IN_ASSERT	FORMATTER / Option to insert a space after the colon in an assert statement - option id: "org.eclipse.jdt.core.formatter.insert_space_after_colon_in_assert" - possible values: { INSERT, DO_NOT_INSERT } - default: INSERT
static final String	FORMATTER_INSERT_SPACE_AFTER_COLON_IN_CASE	FORMATTER / Option to insert a space after colon in a case statement when a opening brace follows the colon - option id: "org.eclipse.jdt.core.formatter.insert_space_after_colon_in_case" - possible values: { INSERT, DO_NOT_INSERT } - default: INSERT
static final String	FORMATTER_INSERT_SPACE_AFTER_COLON_IN_CONDITIONAL	FORMATTER / Option to insert a space after the colon in a conditional expression - option id: "org.eclipse.jdt.core.formatter.insert_space_after_colon_in_conditional" - possible values: { INSERT, DO_NOT_INSERT } - default: INSERT
static final String	FORMATTER_INSERT_SPACE_AFTER_COLON_IN_FOR	FORMATTER / Option to insert a space after colon in a for statement - option id: "org.eclipse.jdt.core.formatter.insert_space_after_colon_in_for" - possible values: { INSERT, DO_NOT_INSERT } - default: INSERT
static final String	FORMATTER_INSERT_SPACE_AFTER_COLON_IN_LABELED_STATEMENT	FORMATTER / Option to insert a space after the colon in a labeled statement - option id: "org.eclipse.jdt.core.formatter.insert_space_after_colon_in_labeled_statem - possible values: { INSERT, DO_NOT_INSERT } - default: INSERT
static final String	FORMATTER_INSERT_SPACE_AFTER_COMMA_IN_ALLOCATION_EXPRESSION	FORMATTER / Option to insert a space after the comma in an allocation expression - option id: "org.eclipse.jdt.core.formatter.insert_space_after_comma_in_allocation_ex - possible values: { INSERT, DO_NOT_INSERT } - default: INSERT
static final String	FORMATTER_INSERT_SPACE_AFTER_COMMA_IN_ANNOTATION	FORMATTER / Option to insert a space after the comma in annotation - option id: "org.eclipse.jdt.core.formatter.insert_space_after_comma_in_annotation" - possible values: { INSERT, DO_NOT_INSERT } - default: INSERT
static final String	FORMATTER_INSERT_SPACE_AFTER_COMMA_IN_ARRAY_INITIALIZER	FORMATTER / Option to insert a space after the comma in an array initializer - option id: "org.eclipse.jdt.core.formatter.insert_space_after_comma_in_array_initiali - possible values: { INSERT, DO_NOT_INSERT } - default: INSERT
static final String	FORMATTER_INSERT_SPACE_AFTER_COMMA_IN_CONSTRUCTOR_DECLARATION_	FORMATTER / Option to insert a space after the comma in the parameters of a constructor declaration - option id: "org.eclipse.jdt.core.formatter.insert_space_after_comma_in_constructor_ - possible values: { INSERT, DO_NOT_INSERT } - default: INSERT

static final String	FORMATTER_INSERT_SPACE_AFTER_COMMA_IN_CONSTRUCTOR_DECLARATION_	FORMATTER / Option to insert a space after the comma in the exception names in a throws clause of a constructor declaration - option id: "org.eclipse.jdt.core.formatter.insert_space_after_comma_in_constructor_d - possible values: { INSERT, DO_NOT_INSERT } - default: INSERT
static final String	FORMATTER_INSERT_SPACE_AFTER_COMMA_IN_ENUM_CONSTANT_ARGUMENTS	FORMATTER / Option to insert a space after the comma in the arguments of an enum constant - option id: "org.eclipse.jdt.core.formatter.insert_space_after_comma_in_enum_consta - possible values: { INSERT, DO_NOT_INSERT } - default: INSERT
static final String	FORMATTER_INSERT_SPACE_AFTER_COMMA_IN_ENUM_DECLARATIONS	FORMATTER / Option to insert a space after the comma in enum declarations - option id: "org.eclipse.jdt.core.formatter.insert_space_after_comma_in_enum_declar - possible values: { INSERT, DO_NOT_INSERT } - default: INSERT
static final String	FORMATTER_INSERT_SPACE_AFTER_COMMA_IN_EXPLICIT_CONSTRUCTOR_CAL	FORMATTER / Option to insert a space after the comma in the arguments of an explicit constructor call - option id: "org.eclipse.jdt.core.formatter.insert_space_after_comma_in_explicitconstr - possible values: { INSERT, DO_NOT_INSERT } - default: INSERT
static final String	FORMATTER_INSERT_SPACE_AFTER_COMMA_IN_FOR_INCREMENTS	FORMATTER / Option to insert a space after the comma in the increments of a for statement - option id: "org.eclipse.jdt.core.formatter.insert_space_after_comma_in_for_incremen - possible values: { INSERT, DO_NOT_INSERT } - default: INSERT
static final String	FORMATTER_INSERT_SPACE_AFTER_COMMA_IN_FOR_INITS	FORMATTER / Option to insert a space after the comma in the initializations of a for statement - option id: "org.eclipse.jdt.core.formatter.insert_space_after_comma_in_for_inits" - possible values: { INSERT, DO_NOT_INSERT } - default: INSERT
static final String	FORMATTER_INSERT_SPACE_AFTER_COMMA_IN_METHOD_DECLARATION_PARAM	FORMATTER / Option to insert a space after the comma in the parameters of a method declaration - option id: "org.eclipse.jdt.core.formatter.insert_space_after_comma_in_method_decl - possible values: { INSERT, DO_NOT_INSERT } - default: INSERT
static final String	FORMATTER_INSERT_SPACE_AFTER_COMMA_IN_METHOD_DECLARATION_THROW	FORMATTER / Option to insert a space after the comma in the exception names in a throws clause of a method declaration - option id: "org.eclipse.jdt.core.formatter.insert_space_after_comma_in_method_decl - possible values: { INSERT, DO_NOT_INSERT } - default: INSERT
static final String	FORMATTER_INSERT_SPACE_AFTER_COMMA_IN_METHOD_INVOCATION_ARGUME	FORMATTER / Option to insert a space after the comma in the arguments of a method invocation - option id: "org.eclipse.jdt.core.formatter.insert_space_after_comma_in_method_invo - possible values: { INSERT, DO_NOT_INSERT } - default: INSERT
static final String	FORMATTER_INSERT_SPACE_AFTER_COMMA_IN_MULTIPLE_FIELD_DECLARATI	FORMATTER / Option to insert a space after the comma in multiple field declaration - option id: "org.eclipse.jdt.core.formatter.insert_space_after_comma_in_multiple_field - possible values: { INSERT, DO_NOT_INSERT } - default: INSERT
static final String	FORMATTER_INSERT_SPACE_AFTER_COMMA_IN_MULTIPLE_LOCAL_DECLARATI	FORMATTER / Option to insert a space after the comma in multiple local declaration - option id: "org.eclipse.jdt.core.formatter.insert_space_after_comma_in_multiple_loca - possible values: { INSERT, DO_NOT_INSERT } - default: INSERT
static final String	FORMATTER_INSERT_SPACE_AFTER_COMMA_IN_PARAMETERIZED_TYPE_REFER	FORMATTER / Option to insert a space after the comma in parameterized type reference - option id: "org.eclipse.jdt.core.formatter.insert_space_after_comma_in_parameterized - possible values: { INSERT, DO_NOT_INSERT } - default: INSERT
static final String	FORMATTER_INSERT_SPACE_AFTER_COMMA_IN_PERMITTED_TYPES	FORMATTER / Option to insert a space after comma in the permitted types list in a type header - option id: "org.eclipse.jdt.core.formatter.insert_space_after_comma_in_permitted_ty - possible values: { INSERT, DO_NOT_INSERT } - default: INSERT
static final String	FORMATTER_INSERT_SPACE_AFTER_COMMA_IN_RECORD_COMPONENTS	FORMATTER / Option to insert a space after comma in record components list - option id: "org.eclipse.jdt.core.formatter.insert_space_after_comma_in_record_comp - possible values: { INSERT, DO_NOT_INSERT } - default: INSERT
static final String	FORMATTER_INSERT_SPACE_AFTER_COMMA_IN_SUPERINTERFACES	FORMATTER / Option to insert a space after the comma in superinterfaces names of a type header - option id: "org.eclipse.jdt.core.formatter.insert_space_after_comma_in_superinterfac - possible values: { INSERT, DO_NOT_INSERT } - default: INSERT
static final String	FORMATTER_INSERT_SPACE_AFTER_COMMA_IN_SWITCH_CASE_EXPRESSIONS	FORMATTER / Option to insert a space after the comma in switch case expressions list - option id: "org.eclipse.jdt.core.formatter.insert_space_after_comma_in_switch_case_ - possible values: { INSERT, DO_NOT_INSERT } - default: INSERT
static final String	FORMATTER_INSERT_SPACE_AFTER_COMMA_IN_TYPE_ARGUMENTS	FORMATTER / Option to insert a space after the comma in explicit type arguments on method/constructor invocations - option id: "org.eclipse.jdt.core.formatter.insert_space_after_comma_in_type_argume - possible values: { INSERT, DO_NOT_INSERT } - default: INSERT

static final String	FORMATTER_INSERT_SPACE_AFTER_COMMA_IN_TYPE_PARAMETERS	FORMATTER / Option to insert a space after the comma in type parameter declarations - option id: "org.eclipse.jdt.core.formatter.insert_space_after_comma_in_type_parame - possible values: { INSERT, DO_NOT_INSERT } - default: INSERT
static final String	FORMATTER_INSERT_SPACE_AFTER_ELLIPSIS	FORMATTER / Option to insert a space after ellipsis - option id: "org.eclipse.jdt.core.formatter.insert_space_after_ellipsis" - possible values: { INSERT, DO_NOT_INSERT } - default: INSERT
static final String	FORMATTER_INSERT_SPACE_AFTER_LAMBDA_ARROW	FORMATTER / Option to insert a space after the -> in lambda expressions - option id: "org.eclipse.jdt.core.formatter.insert_space_after_lambda_arrow" - possible values: { INSERT, DO_NOT_INSERT } - default: INSERT
static final String	FORMATTER_INSERT_SPACE_AFTER_LOGICAL_OPERATOR	FORMATTER / Option to insert a space after a logical operator (&&, ||) - option id: "org.eclipse.jdt.core.formatter.insert_space_after_logical_operator" - possible values: { INSERT, DO_NOT_INSERT } - default: INSERT
static final String	FORMATTER_INSERT_SPACE_AFTER_MULTIPLICATIVE_OPERATOR	FORMATTER / Option to insert a space after a multiplicative operator (*, /, %) - option id: "org.eclipse.jdt.core.formatter.insert_space_after_multiplicative_operator" - possible values: { INSERT, DO_NOT_INSERT } - default: INSERT
static final String	FORMATTER_INSERT_SPACE_AFTER_NOT_OPERATOR	FORMATTER / Option to insert a space after 'not' operator - option id: "org.eclipse.jdt.core.formatter.insert_space_after_not_operator" - possible values: { INSERT, DO_NOT_INSERT } - default: DO_NOT_INSERT
static final String	FORMATTER_INSERT_SPACE_AFTER_OPENING_ANGLE_BRACKET_IN_PARAMETE	FORMATTER / Option to insert a space after the opening angle bracket in parameterized type reference - option id: "org.eclipse.jdt.core.formatter.insert_space_after_opening_angle_bracket_i - possible values: { INSERT, DO_NOT_INSERT } - default: DO_NOT_INSERT
static final String	FORMATTER_INSERT_SPACE_AFTER_OPENING_ANGLE_BRACKET_IN_TYPE_ARG	FORMATTER / Option to insert a space after the opening angle bracket in explicit type arguments on method/constructor invocations - option id: "org.eclipse.jdt.core.formatter.insert_space_after_opening_angle_bracket_i - possible values: { INSERT, DO_NOT_INSERT } - default: DO_NOT_INSERT
static final String	FORMATTER_INSERT_SPACE_AFTER_OPENING_ANGLE_BRACKET_IN_TYPE_PAR	FORMATTER / Option to insert a space after the opening angle bracket in type parameter declarations - option id: "org.eclipse.jdt.core.formatter.insert_space_after_opening_angle_bracket_i - possible values: { INSERT, DO_NOT_INSERT } - default: DO_NOT_INSERT
static final String	FORMATTER_INSERT_SPACE_AFTER_OPENING_BRACE_IN_ARRAY_INITIALIZE	FORMATTER / Option to insert a space after the opening brace in an array initializer - option id: "org.eclipse.jdt.core.formatter.insert_space_after_opening_brace_in_array_ - possible values: { INSERT, DO_NOT_INSERT } - default: DO_NOT_INSERT
static final String	FORMATTER_INSERT_SPACE_AFTER_OPENING_BRACKET_IN_ARRAY_ALLOCATI	FORMATTER / Option to insert a space after the opening bracket inside an array allocation expression - option id: "org.eclipse.jdt.core.formatter.insert_space_after_opening_bracket_in_arra - possible values: { INSERT, DO_NOT_INSERT } - default: DO_NOT_INSERT
static final String	FORMATTER_INSERT_SPACE_AFTER_OPENING_BRACKET_IN_ARRAY_REFERENC	FORMATTER / Option to insert a space after the opening bracket inside an array reference - option id: "org.eclipse.jdt.core.formatter.insert_space_after_opening_bracket_in_arra - possible values: { INSERT, DO_NOT_INSERT } - default: DO_NOT_INSERT
static final String	FORMATTER_INSERT_SPACE_AFTER_OPENING_PAREN_IN_ANNOTATION	FORMATTER / Option to insert a space after the opening parenthesis in annotation - option id: "org.eclipse.jdt.core.formatter.insert_space_after_opening_paren_in_annot - possible values: { INSERT, DO_NOT_INSERT } - default: DO_NOT_INSERT
static final String	FORMATTER_INSERT_SPACE_AFTER_OPENING_PAREN_IN_CAST	FORMATTER / Option to insert a space after the opening parenthesis in a cast expression - option id: "org.eclipse.jdt.core.formatter.insert_space_after_opening_paren_in_cast" - possible values: { INSERT, DO_NOT_INSERT } - default: DO_NOT_INSERT
static final String	FORMATTER_INSERT_SPACE_AFTER_OPENING_PAREN_IN_CATCH	FORMATTER / Option to insert a space after the opening parenthesis in a catch - option id: "org.eclipse.jdt.core.formatter.insert_space_after_opening_paren_in_catch" - possible values: { INSERT, DO_NOT_INSERT } - default: DO_NOT_INSERT

static final String	FORMATTER_INSERT_SPACE_AFTER_OPENING_PAREN_IN_CONSTRUCTOR_DECL	FORMATTER / Option to insert a space after the opening parenthesis in a constructor declaration - option id: "org.eclipse.jdt.core.formatter.insert_space_after_opening_paren_in_const - possible values: { INSERT, DO_NOT_INSERT } - default: DO_NOT_INSERT
static final String	FORMATTER_INSERT_SPACE_AFTER_OPENING_PAREN_IN_ENUM_CONSTANT	FORMATTER / Option to insert a space after the opening parenthesis in enum constant - option id: "org.eclipse.jdt.core.formatter.insert_space_after_opening_paren_in_enum - possible values: { INSERT, DO_NOT_INSERT } - default: DO_NOT_INSERT
static final String	FORMATTER_INSERT_SPACE_AFTER_OPENING_PAREN_IN_FOR	FORMATTER / Option to insert a space after the opening parenthesis in a for statement - option id: "org.eclipse.jdt.core.formatter.insert_space_after_opening_paren_in_for" - possible values: { INSERT, DO_NOT_INSERT } - default: DO_NOT_INSERT
static final String	FORMATTER_INSERT_SPACE_AFTER_OPENING_PAREN_IN_IF	FORMATTER / Option to insert a space after the opening parenthesis in an if statement - option id: "org.eclipse.jdt.core.formatter.insert_space_after_opening_paren_in_if" - possible values: { INSERT, DO_NOT_INSERT } - default: DO_NOT_INSERT
static final String	FORMATTER_INSERT_SPACE_AFTER_OPENING_PAREN_IN_METHOD_DECLARATI	FORMATTER / Option to insert a space after the opening parenthesis in a method declaration - option id: "org.eclipse.jdt.core.formatter.insert_space_after_opening_paren_in_metho - possible values: { INSERT, DO_NOT_INSERT } - default: DO_NOT_INSERT
static final String	FORMATTER_INSERT_SPACE_AFTER_OPENING_PAREN_IN_METHOD_INVOCATIC	FORMATTER / Option to insert a space after the opening parenthesis in a method invocation - option id: "org.eclipse.jdt.core.formatter.insert_space_after_opening_paren_in_metho - possible values: { INSERT, DO_NOT_INSERT } - default: DO_NOT_INSERT
static final String	FORMATTER_INSERT_SPACE_AFTER_OPENING_PAREN_IN_PARENTHESIZED_EX	FORMATTER / Option to insert a space after the opening parenthesis in a parenthesized expression - option id: "org.eclipse.jdt.core.formatter.insert_space_after_opening_paren_in_paren - possible values: { INSERT, DO_NOT_INSERT } - default: DO_NOT_INSERT
static final String	FORMATTER_INSERT_SPACE_AFTER_OPENING_PAREN_IN_RECORD_DECLARATI	FORMATTER / Option to insert a space after the opening parenthesis in a record declaration - option id: "org.eclipse.jdt.core.formatter.insert_space_after_opening_paren_in_record - possible values: { INSERT, DO_NOT_INSERT } - default: DO_NOT_INSERT
static final String	FORMATTER_INSERT_SPACE_AFTER_OPENING_PAREN_IN_SWITCH	FORMATTER / Option to insert a space after the opening parenthesis in a switch statement - option id: "org.eclipse.jdt.core.formatter.insert_space_after_opening_paren_in_switch - possible values: { INSERT, DO_NOT_INSERT } - default: DO_NOT_INSERT
static final String	FORMATTER_INSERT_SPACE_AFTER_OPENING_PAREN_IN_SYNCHRONIZED	FORMATTER / Option to insert a space after the opening parenthesis in a synchronized statement - option id: "org.eclipse.jdt.core.formatter.insert_space_after_opening_paren_in_synch - possible values: { INSERT, DO_NOT_INSERT } - default: DO_NOT_INSERT
static final String	FORMATTER_INSERT_SPACE_AFTER_OPENING_PAREN_IN_TRY	FORMATTER / Option to insert a space after the opening parenthesis in a try with resources statement - option id: "org.eclipse.jdt.core.formatter.insert_space_after_opening_paren_in_try" - possible values: { INSERT, DO_NOT_INSERT } - default: DO_NOT_INSERT
static final String	FORMATTER_INSERT_SPACE_AFTER_OPENING_PAREN_IN_WHILE	FORMATTER / Option to insert a space after the opening parenthesis in a while statement - option id: "org.eclipse.jdt.core.formatter.insert_space_after_opening_paren_in_while - possible values: { INSERT, DO_NOT_INSERT } - default: DO_NOT_INSERT
static final String	FORMATTER_INSERT_SPACE_AFTER_POSTFIX_OPERATOR	FORMATTER / Option to insert a space after a postfix operator - option id: "org.eclipse.jdt.core.formatter.insert_space_after_postfix_operator" - possible values: { INSERT, DO_NOT_INSERT } - default: DO_NOT_INSERT
static final String	FORMATTER_INSERT_SPACE_AFTER_PREFIX_OPERATOR	FORMATTER / Option to insert a space after a prefix operator - option id: "org.eclipse.jdt.core.formatter.insert_space_after_prefix_operator" - possible values: { INSERT, DO_NOT_INSERT } - default: DO_NOT_INSERT

static final String	FORMATTER_INSERT_SPACE_AFTER_QUESTION_IN_CONDITIONAL	FORMATTER / Option to insert a space after question mark in a conditional expression - option id: "org.eclipse.jdt.core.formatter.insert_space_after_question_in_conditional" - possible values: { INSERT, DO_NOT_INSERT } - default: DO_NOT_INSERT
static final String	FORMATTER_INSERT_SPACE_AFTER_QUESTION_IN_WILDCARD	FORMATTER / Option to insert a space after question mark in a wildcard - option id: "org.eclipse.jdt.core.formatter.insert_space_after_question_in_wildcard" - possible values: { INSERT, DO_NOT_INSERT } - default: DO_NOT_INSERT
static final String	FORMATTER_INSERT_SPACE_AFTER_RELATIONAL_OPERATOR	FORMATTER / Option to insert a space after a relational operator (<, >, <=, >=, ==, !=) - option id: "org.eclipse.jdt.core.formatter.insert_space_after_relational_operator" - possible values: { INSERT, DO_NOT_INSERT } - default: INSERT
static final String	FORMATTER_INSERT_SPACE_AFTER_SEMICOLON_IN_FOR	FORMATTER / Option to insert a space after semicolon in a for statement - option id: "org.eclipse.jdt.core.formatter.insert_space_after_semicolon_in_for" - possible values: { INSERT, DO_NOT_INSERT } - default: INSERT
static final String	FORMATTER_INSERT_SPACE_AFTER_SEMICOLON_IN_TRY_RESOURCES	FORMATTER / Option to insert a space after semicolons following each resource declaration in a try with resources statement - option id: "org.eclipse.jdt.core.formatter.insert_space_after_semicolon_in_try_resour - possible values: { INSERT, DO_NOT_INSERT } - default: INSERT
static final String	FORMATTER_INSERT_SPACE_AFTER_SHIFT_OPERATOR	FORMATTER / Option to insert a space after a shift operator (<<, >>, >>>) - option id: "org.eclipse.jdt.core.formatter.insert_space_after_shift_operator" - possible values: { INSERT, DO_NOT_INSERT } - default: INSERT
static final String	FORMATTER_INSERT_SPACE_AFTER_STRING_CONCATENATION	FORMATTER / Option to insert a space after a string concatenation operator - option id: "org.eclipse.jdt.core.formatter.insert_space_after_string_concatenation" - possible values: { INSERT, DO_NOT_INSERT } - default: INSERT
static final String	FORMATTER_INSERT_SPACE_AFTER_UNARY_OPERATOR	FORMATTER / Option to insert a space after an unary operator - option id: "org.eclipse.jdt.core.formatter.insert_space_after_unary_operator" - possible values: { INSERT, DO_NOT_INSERT } - default: DO_NOT_INSERT
static final String	FORMATTER_INSERT_SPACE_BEFORE_ADDITIVE_OPERATOR	FORMATTER / Option to insert a space before an additive operator (+, -) - option id: "org.eclipse.jdt.core.formatter.insert_space_before_additive_operator" - possible values: { INSERT, DO_NOT_INSERT } - default: INSERT
static final String	FORMATTER_INSERT_SPACE_BEFORE_AND_IN_TYPE_PARAMETER	FORMATTER / Option to insert a space before and in wildcard - option id: "org.eclipse.jdt.core.formatter.insert_space_before_and_in_type_paramete - possible values: { INSERT, DO_NOT_INSERT } - default: INSERT
static final String	FORMATTER_INSERT_SPACE_BEFORE_ARROW_IN_SWITCH_CASE	FORMATTER / Option to insert a space before arrow in switch case - option id: "org.eclipse.jdt.core.formatter.insert_space_before_arrow_in_switch_case" - possible values: { INSERT, DO_NOT_INSERT } - default: INSERT
static final String	FORMATTER_INSERT_SPACE_BEFORE_ARROW_IN_SWITCH_DEFAULT	FORMATTER / Option to insert a space before arrow in switch default - option id: "org.eclipse.jdt.core.formatter.insert_space_before_arrow_in_switch_defau - possible values: { INSERT, DO_NOT_INSERT } - default: INSERT
static final String	FORMATTER_INSERT_SPACE_BEFORE_ASSIGNMENT_OPERATOR	FORMATTER / Option to insert a space before an assignment operator - option id: "org.eclipse.jdt.core.formatter.insert_space_before_assignment_operator" - possible values: { INSERT, DO_NOT_INSERT } - default: INSERT
static final String	FORMATTER_INSERT_SPACE_BEFORE_AT_IN_ANNOTATION_TYPE_DECLARATIC	FORMATTER / Option to insert a space before at in annotation type declaration - option id: "org.eclipse.jdt.core.formatter.insert_space_before_at_in_annotation_type_ - possible values: { INSERT, DO_NOT_INSERT } - default: INSERT
static final String	FORMATTER_INSERT_SPACE_BEFORE_BINARY_OPERATOR	**Deprecated.** Use the new settings instead: FORMATTER_INSERT_SPACE_BEFORE_MULTIPLICATIVE_OPERATOR, FORMATTER_INSERT_SPACE_BEFORE_ADDITIVE_OPERATOR, FORMATTER_INSERT_SPACE_BEFORE_STRING_CONCATENATION, FORMATTER_INSERT_SPACE_BEFORE_SHIFT_OPERATOR, FORMATTER_INSERT_SPACE_BEFORE_RELATIONAL_OPERATOR, FORMATTER_INSERT_SPACE_BEFORE_BITWISE_OPERATOR, FORMATTER_INSERT_SPACE_BEFORE_LOGICAL_OPERATOR

static final String	FORMATTER_INSERT_SPACE_BEFORE_BITWISE_OPERATOR	FORMATTER / Option to insert a space before a bitwise operator (&, ^, |) - option id: "org.eclipse.jdt.core.formatter.insert_space_before_bitwise_operator" - possible values: { INSERT, DO_NOT_INSERT } - default: INSERT
static final String	FORMATTER_INSERT_SPACE_BEFORE_CLOSING_ANGLE_BRACKET_IN_PARAMET	FORMATTER / Option to insert a space before the closing angle bracket in parameterized type reference - option id: "org.eclipse.jdt.core.formatter.insert_space_before_closing_angle_bracket_ - possible values: { INSERT, DO_NOT_INSERT } - default: DO_NOT_INSERT
static final String	FORMATTER_INSERT_SPACE_BEFORE_CLOSING_ANGLE_BRACKET_IN_TYPE_AR	FORMATTER / Option to insert a space before the closing angle bracket in explicit type arguments on method/constructor invocations - option id: "org.eclipse.jdt.core.formatter.insert_space_before_closing_angle_bracket_ - possible values: { INSERT, DO_NOT_INSERT } - default: DO_NOT_INSERT
static final String	FORMATTER_INSERT_SPACE_BEFORE_CLOSING_ANGLE_BRACKET_IN_TYPE_PA	FORMATTER / Option to insert a space before the closing angle bracket in type parameter declarations - option id: "org.eclipse.jdt.core.formatter.insert_space_before_closing_angle_bracket_ - possible values: { INSERT, DO_NOT_INSERT } - default: DO_NOT_INSERT
static final String	FORMATTER_INSERT_SPACE_BEFORE_CLOSING_BRACE_IN_ARRAY_INITIALIZ	FORMATTER / Option to insert a space before the closing brace in an array initializer - option id: "org.eclipse.jdt.core.formatter.insert_space_before_closing_brace_in_array - possible values: { INSERT, DO_NOT_INSERT } - default: DO_NOT_INSERT
static final String	FORMATTER_INSERT_SPACE_BEFORE_CLOSING_BRACKET_IN_ARRAY_ALLOCAT	FORMATTER / Option to insert a space before the closing bracket in an array allocation expression - option id: "org.eclipse.jdt.core.formatter.insert_space_before_closing_bracket_in_arra - possible values: { INSERT, DO_NOT_INSERT } - default: DO_NOT_INSERT
static final String	FORMATTER_INSERT_SPACE_BEFORE_CLOSING_BRACKET_IN_ARRAY_REFEREN	FORMATTER / Option to insert a space before the closing bracket in an array reference - option id: "org.eclipse.jdt.core.formatter.insert_space_before_closing_bracket_in_arra - possible values: { INSERT, DO_NOT_INSERT } - default: DO_NOT_INSERT
static final String	FORMATTER_INSERT_SPACE_BEFORE_CLOSING_PAREN_IN_ANNOTATION	FORMATTER / Option to insert a space before the closing parenthesis in annotation - option id: "org.eclipse.jdt.core.formatter.insert_space_before_closing_paren_in_anno - possible values: { INSERT, DO_NOT_INSERT } - default: DO_NOT_INSERT
static final String	FORMATTER_INSERT_SPACE_BEFORE_CLOSING_PAREN_IN_CAST	FORMATTER / Option to insert a space before the closing parenthesis in a cast expression - option id: "org.eclipse.jdt.core.formatter.insert_space_before_closing_paren_in_cast" - possible values: { INSERT, DO_NOT_INSERT } - default: DO_NOT_INSERT
static final String	FORMATTER_INSERT_SPACE_BEFORE_CLOSING_PAREN_IN_CATCH	FORMATTER / Option to insert a space before the closing parenthesis in a catch - option id: "org.eclipse.jdt.core.formatter.insert_space_before_closing_paren_in_catch - possible values: { INSERT, DO_NOT_INSERT } - default: DO_NOT_INSERT
static final String	FORMATTER_INSERT_SPACE_BEFORE_CLOSING_PAREN_IN_CONSTRUCTOR_DEC	FORMATTER / Option to insert a space before the closing parenthesis in a constructor declaration - option id: "org.eclipse.jdt.core.formatter.insert_space_before_closing_paren_in_const - possible values: { INSERT, DO_NOT_INSERT } - default: DO_NOT_INSERT
static final String	FORMATTER_INSERT_SPACE_BEFORE_CLOSING_PAREN_IN_ENUM_CONSTANT	FORMATTER / Option to insert a space before the closing parenthesis in enum constant - option id: "org.eclipse.jdt.core.formatter.insert_space_before_closing_paren_in_enum - possible values: { INSERT, DO_NOT_INSERT } - default: DO_NOT_INSERT
static final String	FORMATTER_INSERT_SPACE_BEFORE_CLOSING_PAREN_IN_FOR	FORMATTER / Option to insert a space before the closing parenthesis in a for statement - option id: "org.eclipse.jdt.core.formatter.insert_space_before_closing_paren_in_for" - possible values: { INSERT, DO_NOT_INSERT } - default: DO_NOT_INSERT
static final String	FORMATTER_INSERT_SPACE_BEFORE_CLOSING_PAREN_IN_IF	FORMATTER / Option to insert a space before the closing parenthesis in an if statement - option id: "org.eclipse.jdt.core.formatter.insert_space_before_closing_paren_in_if" - possible values: { INSERT, DO_NOT_INSERT } - default: DO_NOT_INSERT

static final String	FORMATTER_INSERT_SPACE_BEFORE_CLOSING_PAREN_IN_METHOD_DECLARAT	FORMATTER / Option to insert a space before the closing parenthesis in a method declaration - option id: "org.eclipse.jdt.core.formatter.insert_space_before_closing_paren_in_meth - possible values: { INSERT, DO_NOT_INSERT } - default: DO_NOT_INSERT
static final String	FORMATTER_INSERT_SPACE_BEFORE_CLOSING_PAREN_IN_METHOD_INVOCATI	FORMATTER / Option to insert a space before the closing parenthesis in a method invocation - option id: "org.eclipse.jdt.core.formatter.insert_space_before_closing_paren_in_meth - possible values: { INSERT, DO_NOT_INSERT } - default: DO_NOT_INSERT
static final String	FORMATTER_INSERT_SPACE_BEFORE_CLOSING_PAREN_IN_PARENTHESIZED_E	FORMATTER / Option to insert a space before the closing parenthesis in a parenthesized expression - option id: "org.eclipse.jdt.core.formatter.insert_space_before_closing_paren_in_paren - possible values: { INSERT, DO_NOT_INSERT } - default: DO_NOT_INSERT
static final String	FORMATTER_INSERT_SPACE_BEFORE_CLOSING_PAREN_IN_RECORD_DECLARAT	FORMATTER / Option to insert a space before the closing parenthesis in a record declaration - option id: "org.eclipse.jdt.core.formatter.insert_space_before_closing_paren_in_recor - possible values: { INSERT, DO_NOT_INSERT } - default: DO_NOT_INSERT
static final String	FORMATTER_INSERT_SPACE_BEFORE_CLOSING_PAREN_IN_SWITCH	FORMATTER / Option to insert a space before the closing parenthesis in a switch statement - option id: "org.eclipse.jdt.core.formatter.insert_space_before_closing_paren_in_switc - possible values: { INSERT, DO_NOT_INSERT } - default: DO_NOT_INSERT
static final String	FORMATTER_INSERT_SPACE_BEFORE_CLOSING_PAREN_IN_SYNCHRONIZED	FORMATTER / Option to insert a space before the closing parenthesis in a synchronized statement - option id: "org.eclipse.jdt.core.formatter.insert_space_before_closing_paren_in_synch - possible values: { INSERT, DO_NOT_INSERT } - default: DO_NOT_INSERT
static final String	FORMATTER_INSERT_SPACE_BEFORE_CLOSING_PAREN_IN_TRY	FORMATTER / Option to insert a space before the closing parenthesis in a try with resources statement - option id: "org.eclipse.jdt.core.formatter.insert_space_before_closing_paren_in_try" - possible values: { INSERT, DO_NOT_INSERT } - default: DO_NOT_INSERT
static final String	FORMATTER_INSERT_SPACE_BEFORE_CLOSING_PAREN_IN_WHILE	FORMATTER / Option to insert a space before the closing parenthesis in a while statement - option id: "org.eclipse.jdt.core.formatter.insert_space_before_closing_paren_in_while - possible values: { INSERT, DO_NOT_INSERT } - default: DO_NOT_INSERT
static final String	FORMATTER_INSERT_SPACE_BEFORE_COLON_IN_ASSERT	FORMATTER / Option to insert a space before colon in an assert statement - option id: "org.eclipse.jdt.core.formatter.insert_space_before_colon_in_assert" - possible values: { INSERT, DO_NOT_INSERT } - default: INSERT
static final String	FORMATTER_INSERT_SPACE_BEFORE_COLON_IN_CASE	FORMATTER / Option to insert a space before colon in a case statement - option id: "org.eclipse.jdt.core.formatter.insert_space_before_colon_in_case" - possible values: { INSERT, DO_NOT_INSERT } - default: INSERT
static final String	FORMATTER_INSERT_SPACE_BEFORE_COLON_IN_CONDITIONAL	FORMATTER / Option to insert a space before colon in a conditional expression - option id: "org.eclipse.jdt.core.formatter.insert_space_before_colon_in_conditional" - possible values: { INSERT, DO_NOT_INSERT } - default: INSERT
static final String	FORMATTER_INSERT_SPACE_BEFORE_COLON_IN_DEFAULT	FORMATTER / Option to insert a space before colon in a default statement - option id: "org.eclipse.jdt.core.formatter.insert_space_before_colon_in_default" - possible values: { INSERT, DO_NOT_INSERT } - default: INSERT
static final String	FORMATTER_INSERT_SPACE_BEFORE_COLON_IN_FOR	FORMATTER / Option to insert a space before colon in a for statement - option id: "org.eclipse.jdt.core.formatter.insert_space_before_colon_in_for" - possible values: { INSERT, DO_NOT_INSERT } - default: INSERT
static final String	FORMATTER_INSERT_SPACE_BEFORE_COLON_IN_LABELED_STATEMENT	FORMATTER / Option to insert a space before colon in a labeled statement - option id: "org.eclipse.jdt.core.formatter.insert_space_before_colon_in_labeled_state - possible values: { INSERT, DO_NOT_INSERT } - default: INSERT
static final String	FORMATTER_INSERT_SPACE_BEFORE_COMMA_IN_ALLOCATION_EXPRESSION	FORMATTER / Option to insert a space before comma in an allocation expression - option id: "org.eclipse.jdt.core.formatter.insert_space_before_comma_in_allocation_e - possible values: { INSERT, DO_NOT_INSERT } - default: DO_NOT_INSERT

static final String	FORMATTER_INSERT_SPACE_BEFORE_COMMA_IN_ANNOTATION	FORMATTER / Option to insert a space before comma in annotation - option id: "org.eclipse.jdt.core.formatter.insert_space_before_comma_in_annotation" - possible values: { INSERT, DO_NOT_INSERT } - default: DO_NOT_INSERT
static final String	FORMATTER_INSERT_SPACE_BEFORE_COMMA_IN_ARRAY_INITIALIZER	FORMATTER / Option to insert a space before comma in an array initializer - option id: "org.eclipse.jdt.core.formatter.insert_space_before_comma_in_array_initia - possible values: { INSERT, DO_NOT_INSERT } - default: DO_NOT_INSERT
static final String	FORMATTER_INSERT_SPACE_BEFORE_COMMA_IN_CONSTRUCTOR_DECLARATION	FORMATTER / Option to insert a space before comma in the parameters of a constructor declaration - option id: "org.eclipse.jdt.core.formatter.insert_space_before_comma_in_constructor_ - possible values: { INSERT, DO_NOT_INSERT } - default: DO_NOT_INSERT
static final String	FORMATTER_INSERT_SPACE_BEFORE_COMMA_IN_CONSTRUCTOR_DECLARATION	FORMATTER / Option to insert a space before comma in the exception names of the throws clause of a constructor declaration - option id: "org.eclipse.jdt.core.formatter.insert_space_before_comma_in_constructor_ - possible values: { INSERT, DO_NOT_INSERT } - default: DO_NOT_INSERT
static final String	FORMATTER_INSERT_SPACE_BEFORE_COMMA_IN_ENUM_CONSTANT_ARGUMENTS	FORMATTER / Option to insert a space before comma in the arguments of enum constant - option id: "org.eclipse.jdt.core.formatter.insert_space_before_comma_in_enum_cons - possible values: { INSERT, DO_NOT_INSERT } - default: DO_NOT_INSERT
static final String	FORMATTER_INSERT_SPACE_BEFORE_COMMA_IN_ENUM_DECLARATIONS	FORMATTER / Option to insert a space before comma in enum declarations - option id: "org.eclipse.jdt.core.formatter.insert_space_before_comma_in_enum_decla - possible values: { INSERT, DO_NOT_INSERT } - default: DO_NOT_INSERT
static final String	FORMATTER_INSERT_SPACE_BEFORE_COMMA_IN_EXPLICIT_CONSTRUCTOR_CA	FORMATTER / Option to insert a space before comma in the arguments of an explicit constructor call - option id: "org.eclipse.jdt.core.formatter.insert_space_before_comma_in_explicitcons - possible values: { INSERT, DO_NOT_INSERT } - default: DO_NOT_INSERT
static final String	FORMATTER_INSERT_SPACE_BEFORE_COMMA_IN_FOR_INCREMENTS	FORMATTER / Option to insert a space before comma in the increments of a for statement - option id: "org.eclipse.jdt.core.formatter.insert_space_before_comma_in_for_increme - possible values: { INSERT, DO_NOT_INSERT } - default: DO_NOT_INSERT
static final String	FORMATTER_INSERT_SPACE_BEFORE_COMMA_IN_FOR_INITS	FORMATTER / Option to insert a space before comma in the initializations of a for statement - option id: "org.eclipse.jdt.core.formatter.insert_space_before_comma_in_for_inits" - possible values: { INSERT, DO_NOT_INSERT } - default: DO_NOT_INSERT
static final String	FORMATTER_INSERT_SPACE_BEFORE_COMMA_IN_METHOD_DECLARATION_PARA	FORMATTER / Option to insert a space before comma in the parameters of a method declaration - option id: "org.eclipse.jdt.core.formatter.insert_space_before_comma_in_method_dec - possible values: { INSERT, DO_NOT_INSERT } - default: DO_NOT_INSERT
static final String	FORMATTER_INSERT_SPACE_BEFORE_COMMA_IN_METHOD_DECLARATION_THRO	FORMATTER / Option to insert a space before comma in the exception names of the throws clause of a method declaration - option id: "org.eclipse.jdt.core.formatter.insert_space_before_comma_in_method_dec - possible values: { INSERT, DO_NOT_INSERT } - default: DO_NOT_INSERT
static final String	FORMATTER_INSERT_SPACE_BEFORE_COMMA_IN_METHOD_INVOCATION_ARGUM	FORMATTER / Option to insert a space before comma in the arguments of a method invocation - option id: "org.eclipse.jdt.core.formatter.insert_space_before_comma_in_method_inv - possible values: { INSERT, DO_NOT_INSERT } - default: DO_NOT_INSERT
static final String	FORMATTER_INSERT_SPACE_BEFORE_COMMA_IN_MULTIPLE_FIELD_DECLARAT	FORMATTER / Option to insert a space before comma in a multiple field declaration - option id: "org.eclipse.jdt.core.formatter.insert_space_before_comma_in_multiple_fie - possible values: { INSERT, DO_NOT_INSERT } - default: DO_NOT_INSERT
static final String	FORMATTER_INSERT_SPACE_BEFORE_COMMA_IN_MULTIPLE_LOCAL_DECLARAT	FORMATTER / Option to insert a space before comma in a multiple local declaration - option id: "org.eclipse.jdt.core.formatter.insert_space_before_comma_in_multiple_lo - possible values: { INSERT, DO_NOT_INSERT } - default: DO_NOT_INSERT

static final String	FORMATTER_INSERT_SPACE_BEFORE_COMMA_IN_PARAMETERIZED_TYPE_REFE	FORMATTER / Option to insert a space before comma in parameterized type reference - option id: "org.eclipse.jdt.core.formatter.insert_space_before_comma_in_parameteriz - possible values: { INSERT, DO_NOT_INSERT } - default: DO_NOT_INSERT
static final String	FORMATTER_INSERT_SPACE_BEFORE_COMMA_IN_PERMITTED_TYPES	FORMATTER / Option to insert a space before comma in the permitted types list in a type header - option id: "org.eclipse.jdt.core.formatter.insert_space_before_comma_in_permitted_t - possible values: { INSERT, DO_NOT_INSERT } - default: DO_NOT_INSERT
static final String	FORMATTER_INSERT_SPACE_BEFORE_COMMA_IN_RECORD_COMPONENTS	FORMATTER / Option to insert a space before comma in record components list - option id: "org.eclipse.jdt.core.formatter.insert_space_before_comma_in_record_com - possible values: { INSERT, DO_NOT_INSERT } - default: DO_NOT_INSERT
static final String	FORMATTER_INSERT_SPACE_BEFORE_COMMA_IN_SUPERINTERFACES	FORMATTER / Option to insert a space before comma in the superinterfaces names in a type header - option id: "org.eclipse.jdt.core.formatter.insert_space_before_comma_in_superinterfa - possible values: { INSERT, DO_NOT_INSERT } - default: DO_NOT_INSERT
static final String	FORMATTER_INSERT_SPACE_BEFORE_COMMA_IN_SWITCH_CASE_EXPRESSIONS	FORMATTER / Option to insert a space before the comma in switch case expressions list - option id: "org.eclipse.jdt.core.formatter.insert_space_before_comma_in_switch_case - possible values: { INSERT, DO_NOT_INSERT } - default: DO_NOT_INSERT
static final String	FORMATTER_INSERT_SPACE_BEFORE_COMMA_IN_TYPE_ARGUMENTS	FORMATTER / Option to insert a space before comma in explicit type arguments on method/constructor invocations - option id: "org.eclipse.jdt.core.formatter.insert_space_before_comma_in_type_argum - possible values: { INSERT, DO_NOT_INSERT } - default: DO_NOT_INSERT
static final String	FORMATTER_INSERT_SPACE_BEFORE_COMMA_IN_TYPE_PARAMETERS	FORMATTER / Option to insert a space before comma in type parameter declarations - option id: "org.eclipse.jdt.core.formatter.insert_space_before_comma_in_type_param - possible values: { INSERT, DO_NOT_INSERT } - default: DO_NOT_INSERT
static final String	FORMATTER_INSERT_SPACE_BEFORE_ELLIPSIS	FORMATTER / Option to insert a space before ellipsis - option id: "org.eclipse.jdt.core.formatter.insert_space_before_ellipsis" - possible values: { INSERT, DO_NOT_INSERT } - default: DO_NOT_INSERT
static final String	FORMATTER_INSERT_SPACE_BEFORE_LAMBDA_ARROW	FORMATTER / Option to insert a space before lambda -> - option id: "org.eclipse.jdt.core.formatter.insert_space_before_lambda_arrow" - possible values: { INSERT, DO_NOT_INSERT } - default: INSERT
static final String	FORMATTER_INSERT_SPACE_BEFORE_LOGICAL_OPERATOR	FORMATTER / Option to insert a space before a logical operator (&&, ||) - option id: "org.eclipse.jdt.core.formatter.insert_space_before_logical_operator" - possible values: { INSERT, DO_NOT_INSERT } - default: INSERT
static final String	FORMATTER_INSERT_SPACE_BEFORE_MULTIPLICATIVE_OPERATOR	FORMATTER / Option to insert a space before a multiplicative operator (*, /, %) - option id: "org.eclipse.jdt.core.formatter.insert_space_before_multiplicative_operator" - possible values: { INSERT, DO_NOT_INSERT } - default: INSERT
static final String	FORMATTER_INSERT_SPACE_BEFORE_OPENING_ANGLE_BRACKET_IN_PARAMET	FORMATTER / Option to insert a space before the opening angle bracket in parameterized type reference - option id: "org.eclipse.jdt.core.formatter.insert_space_before_opening_angle_bracket - possible values: { INSERT, DO_NOT_INSERT } - default: DO_NOT_INSERT
static final String	FORMATTER_INSERT_SPACE_BEFORE_OPENING_ANGLE_BRACKET_IN_TYPE_AR	FORMATTER / Option to insert a space before the opening angle bracket in explicit type arguments on method/constructor invocations - option id: "org.eclipse.jdt.core.formatter.insert_space_before_opening_angle_bracket - possible values: { INSERT, DO_NOT_INSERT } - default: DO_NOT_INSERT
static final String	FORMATTER_INSERT_SPACE_BEFORE_OPENING_ANGLE_BRACKET_IN_TYPE_PA	FORMATTER / Option to insert a space before the opening angle bracket in type parameter declarations - option id: "org.eclipse.jdt.core.formatter.insert_space_before_opening_angle_bracket - possible values: { INSERT, DO_NOT_INSERT } - default: DO_NOT_INSERT
static final String	FORMATTER_INSERT_SPACE_BEFORE_OPENING_BRACE_IN_ANNOTATION_TYPE	FORMATTER / Option to insert a space before the opening brace in an annotation type declaration - option id: "org.eclipse.jdt.core.formatter.insert_space_before_opening_brace_in_anno - possible values: { INSERT, DO_NOT_INSERT } - default: INSERT

static final String	FORMATTER_INSERT_SPACE_BEFORE_OPENING_BRACE_IN_ANONYMOUS_TYPE_	FORMATTER / Option to insert a space before the opening brace in an anonymous type declaration - option id: "org.eclipse.jdt.core.formatter.insert_space_before_opening_brace_in_anor - possible values: { INSERT, DO_NOT_INSERT } - default: INSERT
static final String	FORMATTER_INSERT_SPACE_BEFORE_OPENING_BRACE_IN_ARRAY_INITIALIZ	FORMATTER / Option to insert a space before the opening brace in an array initializer - option id: "org.eclipse.jdt.core.formatter.insert_space_before_opening_brace_in_arra - possible values: { INSERT, DO_NOT_INSERT } - default: DO_NOT_INSERT
static final String	FORMATTER_INSERT_SPACE_BEFORE_OPENING_BRACE_IN_BLOCK	FORMATTER / Option to insert a space before the opening brace in a block - option id: "org.eclipse.jdt.core.formatter.insert_space_before_opening_brace_in_bloc - possible values: { INSERT, DO_NOT_INSERT } - default: INSERT
static final String	FORMATTER_INSERT_SPACE_BEFORE_OPENING_BRACE_IN_CONSTRUCTOR_DEC	FORMATTER / Option to insert a space before the opening brace in a constructor declaration - option id: "org.eclipse.jdt.core.formatter.insert_space_before_opening_brace_in_cons - possible values: { INSERT, DO_NOT_INSERT } - default: INSERT
static final String	FORMATTER_INSERT_SPACE_BEFORE_OPENING_BRACE_IN_ENUM_CONSTANT	FORMATTER / Option to insert a space before the opening brace in an enum constant - option id: "org.eclipse.jdt.core.formatter.insert_space_before_opening_brace_in_enur - possible values: { INSERT, DO_NOT_INSERT } - default: INSERT
static final String	FORMATTER_INSERT_SPACE_BEFORE_OPENING_BRACE_IN_ENUM_DECLARATIC	FORMATTER / Option to insert a space before the opening brace in an enum declaration - option id: "org.eclipse.jdt.core.formatter.insert_space_before_opening_brace_in_enur - possible values: { INSERT, DO_NOT_INSERT } - default: INSERT
static final String	FORMATTER_INSERT_SPACE_BEFORE_OPENING_BRACE_IN_METHOD_DECLARAT	FORMATTER / Option to insert a space before the opening brace in a method declaration - option id: "org.eclipse.jdt.core.formatter.insert_space_before_opening_brace_in_metl - possible values: { INSERT, DO_NOT_INSERT } - default: INSERT
static final String	FORMATTER_INSERT_SPACE_BEFORE_OPENING_BRACE_IN_RECORD_CONSTRUC	FORMATTER / Option to insert a space before the opening brace in a record constructor - option id: "org.eclipse.jdt.core.formatter.insert_space_before_opening_brace_in_reco - possible values: { INSERT, DO_NOT_INSERT } - default: INSERT
static final String	FORMATTER_INSERT_SPACE_BEFORE_OPENING_BRACE_IN_RECORD_DECLARAT	FORMATTER / Option to insert a space before the opening brace in a record declaration - option id: "org.eclipse.jdt.core.formatter.insert_space_before_opening_brace_in_reco - possible values: { INSERT, DO_NOT_INSERT } - default: INSERT
static final String	FORMATTER_INSERT_SPACE_BEFORE_OPENING_BRACE_IN_SWITCH	FORMATTER / Option to insert a space before the opening brace in a switch statement - option id: "org.eclipse.jdt.core.formatter.insert_space_before_opening_brace_in_swit - possible values: { INSERT, DO_NOT_INSERT } - default: INSERT
static final String	FORMATTER_INSERT_SPACE_BEFORE_OPENING_BRACE_IN_TYPE_DECLARATIC	FORMATTER / Option to insert a space before the opening brace in a type declaration - option id: "org.eclipse.jdt.core.formatter.insert_space_before_opening_brace_in_type - possible values: { INSERT, DO_NOT_INSERT } - default: INSERT
static final String	FORMATTER_INSERT_SPACE_BEFORE_OPENING_BRACKET_IN_ARRAY_ALLOCAT	FORMATTER / Option to insert a space before the opening bracket in an array allocation expression - option id: "org.eclipse.jdt.core.formatter.insert_space_before_opening_bracket_in_ar - possible values: { INSERT, DO_NOT_INSERT } - default: DO_NOT_INSERT
static final String	FORMATTER_INSERT_SPACE_BEFORE_OPENING_BRACKET_IN_ARRAY_REFEREN	FORMATTER / Option to insert a space before the opening bracket in an array reference - option id: "org.eclipse.jdt.core.formatter.insert_space_before_opening_bracket_in_ar - possible values: { INSERT, DO_NOT_INSERT } - default: DO_NOT_INSERT
static final String	FORMATTER_INSERT_SPACE_BEFORE_OPENING_BRACKET_IN_ARRAY_TYPE_RE	FORMATTER / Option to insert a space before the opening bracket in an array type reference - option id: "org.eclipse.jdt.core.formatter.insert_space_before_opening_bracket_in_ar - possible values: { INSERT, DO_NOT_INSERT } - default: DO_NOT_INSERT
static final String	FORMATTER_INSERT_SPACE_BEFORE_OPENING_PAREN_IN_ANNOTATION	FORMATTER / Option to insert a space before the opening parenthesis in annotation - option id: "org.eclipse.jdt.core.formatter.insert_space_before_opening_paren_in_ann - possible values: { INSERT, DO_NOT_INSERT } - default: DO_NOT_INSERT

static final String	FORMATTER_INSERT_SPACE_BEFORE_OPENING_PAREN_IN_ANNOTATION_TYPE	FORMATTER / Option to insert a space before the opening parenthesis in annotation type member declaration - option id: "org.eclipse.jdt.core.formatter.insert_space_before_opening_paren_in_ann - possible values: { INSERT, DO_NOT_INSERT } - default: DO_NOT_INSERT
static final String	FORMATTER_INSERT_SPACE_BEFORE_OPENING_PAREN_IN_CATCH	FORMATTER / Option to insert a space before the opening parenthesis in a catch - option id: "org.eclipse.jdt.core.formatter.insert_space_before_opening_paren_in_catc - possible values: { INSERT, DO_NOT_INSERT } - default: INSERT
static final String	FORMATTER_INSERT_SPACE_BEFORE_OPENING_PAREN_IN_CONSTRUCTOR_DEC	FORMATTER / Option to insert a space before the opening parenthesis in a constructor declaration - option id: "org.eclipse.jdt.core.formatter.insert_space_before_opening_paren_in_cons - possible values: { INSERT, DO_NOT_INSERT } - default: DO_NOT_INSERT
static final String	FORMATTER_INSERT_SPACE_BEFORE_OPENING_PAREN_IN_ENUM_CONSTANT	FORMATTER / Option to insert a space before the opening parenthesis in enum constant - option id: "org.eclipse.jdt.core.formatter.insert_space_before_opening_paren_in_enu - possible values: { INSERT, DO_NOT_INSERT } - default: DO_NOT_INSERT
static final String	FORMATTER_INSERT_SPACE_BEFORE_OPENING_PAREN_IN_FOR	FORMATTER / Option to insert a space before the opening parenthesis in a for statement - option id: "org.eclipse.jdt.core.formatter.insert_space_before_opening_paren_in_for" - possible values: { INSERT, DO_NOT_INSERT } - default: INSERT
static final String	FORMATTER_INSERT_SPACE_BEFORE_OPENING_PAREN_IN_IF	FORMATTER / Option to insert a space before the opening parenthesis in an if statement - option id: "org.eclipse.jdt.core.formatter.insert_space_before_opening_paren_in_if" - possible values: { INSERT, DO_NOT_INSERT } - default: INSERT
static final String	FORMATTER_INSERT_SPACE_BEFORE_OPENING_PAREN_IN_METHOD_DECLARAT	FORMATTER / Option to insert a space before the opening parenthesis in a method declaration - option id: "org.eclipse.jdt.core.formatter.insert_space_before_opening_paren_in_met - possible values: { INSERT, DO_NOT_INSERT } - default: DO_NOT_INSERT
static final String	FORMATTER_INSERT_SPACE_BEFORE_OPENING_PAREN_IN_METHOD_INVOCATI	FORMATTER / Option to insert a space before the opening parenthesis in a method invocation - option id: "org.eclipse.jdt.core.formatter.insert_space_before_opening_paren_in_met - possible values: { INSERT, DO_NOT_INSERT } - default: DO_NOT_INSERT
static final String	FORMATTER_INSERT_SPACE_BEFORE_OPENING_PAREN_IN_PARENTHESIZED_E	FORMATTER / Option to insert a space before the opening parenthesis in a parenthesized expression - option id: "org.eclipse.jdt.core.formatter.insert_space_before_opening_paren_in_pare - possible values: { INSERT, DO_NOT_INSERT } - default: DO_NOT_INSERT
static final String	FORMATTER_INSERT_SPACE_BEFORE_OPENING_PAREN_IN_RECORD_DECLARAT	FORMATTER / Option to insert a space before the opening parenthesis in a record declaration - option id: "org.eclipse.jdt.core.formatter.insert_space_before_opening_paren_in_reco - possible values: { INSERT, DO_NOT_INSERT } - default: DO_NOT_INSERT
static final String	FORMATTER_INSERT_SPACE_BEFORE_OPENING_PAREN_IN_SWITCH	FORMATTER / Option to insert a space before the opening parenthesis in a switch statement - option id: "org.eclipse.jdt.core.formatter.insert_space_before_opening_paren_in_swit - possible values: { INSERT, DO_NOT_INSERT } - default: INSERT
static final String	FORMATTER_INSERT_SPACE_BEFORE_OPENING_PAREN_IN_SYNCHRONIZED	FORMATTER / Option to insert a space before the opening parenthesis in a synchronized statement - option id: "org.eclipse.jdt.core.formatter.insert_space_before_opening_paren_in_sync - possible values: { INSERT, DO_NOT_INSERT } - default: INSERT
static final String	FORMATTER_INSERT_SPACE_BEFORE_OPENING_PAREN_IN_TRY	FORMATTER / Option to insert a space before the opening parenthesis in a try with resources statement - option id: "org.eclipse.jdt.core.formatter.insert_space_before_opening_paren_in_try" - possible values: { INSERT, DO_NOT_INSERT } - default: INSERT
static final String	FORMATTER_INSERT_SPACE_BEFORE_OPENING_PAREN_IN_WHILE	FORMATTER / Option to insert a space before the opening parenthesis in a while statement - option id: "org.eclipse.jdt.core.formatter.insert_space_before_opening_paren_in_whil - possible values: { INSERT, DO_NOT_INSERT } - default: INSERT
static final String	FORMATTER_INSERT_SPACE_BEFORE_PARENTHESIZED_EXPRESSION_IN_RETU	FORMATTER / Option to insert a space before parenthesized expression in return statement - option id: "org.eclipse.jdt.core.formatter.insert_space_before_parenthesized_expressio - possible values: { INSERT, DO_NOT_INSERT } - default: INSERT

static final String	FORMATTER_INSERT_SPACE_BEFORE_PARENTHESIZED_EXPRESSION_IN_THRO	FORMATTER / Option to insert a space before parenthesized expression in throw statement - option id: "org.eclipse.jdt.core.formatter.insert_space_before_parenthesized_expressio - possible values: { INSERT, DO_NOT_INSERT } - default: INSERT
static final String	FORMATTER_INSERT_SPACE_BEFORE_POSTFIX_OPERATOR	FORMATTER / Option to insert a space before a postfix operator - option id: "org.eclipse.jdt.core.formatter.insert_space_before_postfix_operator" - possible values: { INSERT, DO_NOT_INSERT } - default: DO_NOT_INSERT
static final String	FORMATTER_INSERT_SPACE_BEFORE_PREFIX_OPERATOR	FORMATTER / Option to insert a space before a prefix operator - option id: "org.eclipse.jdt.core.formatter.insert_space_before_prefix_operator" - possible values: { INSERT, DO_NOT_INSERT } - default: DO_NOT_INSERT
static final String	FORMATTER_INSERT_SPACE_BEFORE_QUESTION_IN_CONDITIONAL	FORMATTER / Option to insert a space before question mark in a conditional expression - option id: "org.eclipse.jdt.core.formatter.insert_space_before_question_in_conditiona - possible values: { INSERT, DO_NOT_INSERT } - default: INSERT
static final String	FORMATTER_INSERT_SPACE_BEFORE_QUESTION_IN_WILDCARD	FORMATTER / Option to insert a space before question mark in a wildcard - option id: "org.eclipse.jdt.core.formatter.insert_space_before_question_in_wildcard" - possible values: { INSERT, DO_NOT_INSERT } - default: DO_NOT_INSERT
static final String	FORMATTER_INSERT_SPACE_BEFORE_RELATIONAL_OPERATOR	FORMATTER / Option to insert a space before a relational operator (<, >, <=, >=, ==, !=) - option id: "org.eclipse.jdt.core.formatter.insert_space_before_relational_operator" - possible values: { INSERT, DO_NOT_INSERT } - default: INSERT
static final String	FORMATTER_INSERT_SPACE_BEFORE_SEMICOLON	FORMATTER / Option to insert a space before semicolon - option id: "org.eclipse.jdt.core.formatter.insert_space_before_semicolon" - possible values: { INSERT, DO_NOT_INSERT } - default: DO_NOT_INSERT
static final String	FORMATTER_INSERT_SPACE_BEFORE_SEMICOLON_IN_FOR	FORMATTER / Option to insert a space before semicolon in for statement - option id: "org.eclipse.jdt.core.formatter.insert_space_before_semicolon_in_for" - possible values: { INSERT, DO_NOT_INSERT } - default: DO_NOT_INSERT
static final String	FORMATTER_INSERT_SPACE_BEFORE_SEMICOLON_IN_TRY_RESOURCES	FORMATTER / Option to insert a space before semicolons following each resource declaration in a try with resources statement - option id: "org.eclipse.jdt.core.formatter.insert_space_before_semicolon_in_try_resou - possible values: { INSERT, DO_NOT_INSERT } - default: DO_NOT_INSERT
static final String	FORMATTER_INSERT_SPACE_BEFORE_SHIFT_OPERATOR	FORMATTER / Option to insert a space before a shift operator (<<, >>, >>>) - option id: "org.eclipse.jdt.core.formatter.insert_space_before_shift_operator" - possible values: { INSERT, DO_NOT_INSERT } - default: INSERT
static final String	FORMATTER_INSERT_SPACE_BEFORE_STRING_CONCATENATION	FORMATTER / Option to insert a space before a string concatenation operator - option id: "org.eclipse.jdt.core.formatter.insert_space_before_string_concatenation" - possible values: { INSERT, DO_NOT_INSERT } - default: INSERT
static final String	FORMATTER_INSERT_SPACE_BEFORE_UNARY_OPERATOR	FORMATTER / Option to insert a space before unary operator - option id: "org.eclipse.jdt.core.formatter.insert_space_before_unary_operator" - possible values: { INSERT, DO_NOT_INSERT } - default: DO_NOT_INSERT
static final String	FORMATTER_INSERT_SPACE_BETWEEN_BRACKETS_IN_ARRAY_TYPE_REFERENC	FORMATTER / Option to insert a space between brackets in an array type reference - option id: "org.eclipse.jdt.core.formatter.insert_space_between_brackets_in_array_ty - possible values: { INSERT, DO_NOT_INSERT } - default: DO_NOT_INSERT
static final String	FORMATTER_INSERT_SPACE_BETWEEN_EMPTY_BRACES_IN_ARRAY_INITIALIZ	FORMATTER / Option to insert a space between empty braces in an array initializer - option id: "org.eclipse.jdt.core.formatter.insert_space_between_empty_braces_in_arr - possible values: { INSERT, DO_NOT_INSERT } - default: DO_NOT_INSERT
static final String	FORMATTER_INSERT_SPACE_BETWEEN_EMPTY_BRACKETS_IN_ARRAY_ALLOCAT	FORMATTER / Option to insert a space between empty brackets in an array allocation expression - option id: "org.eclipse.jdt.core.formatter.insert_space_between_empty_brackets_in_a - possible values: { INSERT, DO_NOT_INSERT } - default: DO_NOT_INSERT

static final String	FORMATTER_INSERT_SPACE_BETWEEN_EMPTY_PARENS_IN_ANNOTATION_TYPE	FORMATTER / Option to insert a space between empty parenthesis in an annotation type member declaration - option id: "org.eclipse.jdt.core.formatter.insert_space_between_empty_parens_in_an - possible values: { INSERT, DO_NOT_INSERT } - default: DO_NOT_INSERT
static final String	FORMATTER_INSERT_SPACE_BETWEEN_EMPTY_PARENS_IN_CONSTRUCTOR_DEC	FORMATTER / Option to insert a space between empty parenthesis in a constructor declaration - option id: "org.eclipse.jdt.core.formatter.insert_space_between_empty_parens_in_co - possible values: { INSERT, DO_NOT_INSERT } - default: DO_NOT_INSERT
static final String	FORMATTER_INSERT_SPACE_BETWEEN_EMPTY_PARENS_IN_ENUM_CONSTANT	FORMATTER / Option to insert a space between empty parenthesis in enum constant - option id: "org.eclipse.jdt.core.formatter.insert_space_between_empty_parens_in_en - possible values: { INSERT, DO_NOT_INSERT } - default: DO_NOT_INSERT
static final String	FORMATTER_INSERT_SPACE_BETWEEN_EMPTY_PARENS_IN_METHOD_DECLARAT	FORMATTER / Option to insert a space between empty parenthesis in a method declaration - option id: "org.eclipse.jdt.core.formatter.insert_space_between_empty_parens_in_me - possible values: { INSERT, DO_NOT_INSERT } - default: DO_NOT_INSERT
static final String	FORMATTER_INSERT_SPACE_BETWEEN_EMPTY_PARENS_IN_METHOD_INVOCATI	FORMATTER / Option to insert a space between empty parenthesis in a method invocation - option id: "org.eclipse.jdt.core.formatter.insert_space_between_empty_parens_in_me - possible values: { INSERT, DO_NOT_INSERT } - default: DO_NOT_INSERT
static final String	FORMATTER_JOIN_LINE_COMMENTS	FORMATTER / Option to specify whether the formatter can join consecutive line comments - option id: "org.eclipse.jdt.core.formatter.join_line_comments" - possible values: { TRUE, FALSE } - default: FALSE
static final String	FORMATTER_JOIN_LINES_IN_COMMENTS	FORMATTER / Option to specify whether the formatter can join text lines in comments or not For example, the following comment: /** * The foo method
static final String	FORMATTER_JOIN_WRAPPED_LINES	FORMATTER / Option to specify whether the formatter can join wrapped lines or not For example, the wrapped lines of method foo return statement in following test case: class X { String foo() { return "select x " + "from y " + "where z=a"; } } will be preserved by the formatter when the new preference is used even if the maximum line width would give it enough space to join the lines.
static final String	FORMATTER_KEEP_ANNOTATION_DECLARATION_ON_ONE_LINE	FORMATTER / Option to control when an annotation declaration should be kept on one line - option id: "org.eclipse.jdt.core.formatter.keep_annotation_declaration_on_one_line" - possible values: { ONE_LINE_NEVER, ONE_LINE_IF_EMPTY, ONE_LINE_IF_SINGLE_ITEM, ONE_LINE_ALWAYS, ONE_LINE_PRESERVE } - default: ONE_LINE_NEVER
static final String	FORMATTER_KEEP_ANONYMOUS_TYPE_DECLARATION_ON_ONE_LINE	FORMATTER / Option to control when an anonymous type declaration should be kept on one line - option id: "org.eclipse.jdt.core.formatter.keep_anonymous_type_declaration_on_one_ - possible values: { ONE_LINE_NEVER, ONE_LINE_IF_EMPTY, ONE_LINE_IF_SINGLE_ITEM, ONE_LINE_ALWAYS, ONE_LINE_PRESERVE } - default: ONE_LINE_NEVER
static final String	FORMATTER_KEEP_CODE_BLOCK_ON_ONE_LINE	FORMATTER / Option to control when a code block other than if-then and loop body should be kept on one line - option id: "org.eclipse.jdt.core.formatter.keep_code_block_on_one_line" - possible values: { ONE_LINE_NEVER, ONE_LINE_IF_EMPTY } - default: ONE_LINE_NEVER
static final String	FORMATTER_KEEP_ELSE_STATEMENT_ON_SAME_LINE	FORMATTER / Option to keep else statement on the same line - option id: "org.eclipse.jdt.core.formatter.keep_else_statement_on_same_line" - possible values: { TRUE, FALSE } - default: FALSE
static final String	FORMATTER_KEEP_EMPTY_ARRAY_INITIALIZER_ON_ONE_LINE	FORMATTER / Option to keep empty array initializer one one line - option id: "org.eclipse.jdt.core.formatter.keep_empty_array_initializer_on_one_line" - possible values: { TRUE, FALSE } - default: FALSE
static final String	FORMATTER_KEEP_ENUM_CONSTANT_DECLARATION_ON_ONE_LINE	FORMATTER / Option to control when an enum constant declaration body should be kept on one line - option id: "org.eclipse.jdt.core.formatter.keep_enum_constant_declaration_on_one_li - possible values: { ONE_LINE_NEVER, ONE_LINE_IF_EMPTY, ONE_LINE_IF_SINGLE_ITEM, ONE_LINE_ALWAYS, ONE_LINE_PRESERVE } - default: ONE_LINE_NEVER

static final String	FORMATTER_KEEP_ENUM_DECLARATION_ON_ONE_LINE	FORMATTER / Option to control when an enum declaration should be kept on one line - option id: "org.eclipse.jdt.core.formatter.keep_enum_declaration_on_one_line" - possible values: { ONE_LINE_NEVER, ONE_LINE_IF_EMPTY, ONE_LINE_IF_SINGLE_ITEM, ONE_LINE_ALWAYS, ONE_LINE_PRESERVE } - default: ONE_LINE_NEVER
static final String	FORMATTER_KEEP_GUARDIAN_CLAUSE_ON_ONE_LINE	FORMATTER / Option to keep guardian clause on one line, in addition to the #FORMATTER_KEEP_IF_THEN_BODY_BLOCK_ON_ONE_LINE option - option id: "org.eclipse.jdt.core.formatter.format_guardian_clause_on_one_line" - possible values: { TRUE, FALSE } - default: FALSE
static final String	FORMATTER_KEEP_IF_THEN_BODY_BLOCK_ON_ONE_LINE	FORMATTER / Option to control when an if-then statement body block should be kept on one line - option id: "org.eclipse.jdt.core.formatter.keep_if_then_body_block_on_one_line" - possible values: { ONE_LINE_NEVER, ONE_LINE_IF_EMPTY, ONE_LINE_IF_SINGLE_ITEM, ONE_LINE_ALWAYS, ONE_LINE_PRESERVE } - default: ONE_LINE_NEVER
static final String	FORMATTER_KEEP_LAMBDA_BODY_BLOCK_ON_ONE_LINE	FORMATTER / Option to control when a lambda body should be kept on one line - option id: "org.eclipse.jdt.core.formatter.keep_lambda_body_block_on_one_line" - possible values: { ONE_LINE_NEVER, ONE_LINE_IF_EMPTY, ONE_LINE_IF_SINGLE_ITEM, ONE_LINE_ALWAYS, ONE_LINE_PRESERVE } - default: ONE_LINE_NEVER
static final String	FORMATTER_KEEP_LOOP_BODY_BLOCK_ON_ONE_LINE	FORMATTER / Option to control when a loop body block should be kept on one line - option id: "org.eclipse.jdt.core.formatter.keep_loop_body_block_on_one_line" - possible values: { ONE_LINE_NEVER, ONE_LINE_IF_EMPTY, ONE_LINE_IF_SINGLE_ITEM, ONE_LINE_ALWAYS, ONE_LINE_PRESERVE } - default: ONE_LINE_NEVER
static final String	FORMATTER_KEEP_METHOD_BODY_ON_ONE_LINE	FORMATTER / Option to control when a method body should be kept on one line - option id: "org.eclipse.jdt.core.formatter.keep_method_body_on_one_line" - possible values: { ONE_LINE_NEVER, ONE_LINE_IF_EMPTY, ONE_LINE_IF_SINGLE_ITEM, ONE_LINE_ALWAYS, ONE_LINE_PRESERVE } - default: ONE_LINE_NEVER
static final String	FORMATTER_KEEP_RECORD_CONSTRUCTOR_ON_ONE_LINE	FORMATTER / Option to control when a record constructor should be kept on one line - option id: "org.eclipse.jdt.core.formatter.keep_record_constructor_on_one_line" - possible values: { ONE_LINE_NEVER, ONE_LINE_IF_EMPTY, ONE_LINE_IF_SINGLE_ITEM, ONE_LINE_ALWAYS, ONE_LINE_PRESERVE } - default: ONE_LINE_NEVER
static final String	FORMATTER_KEEP_RECORD_DECLARATION_ON_ONE_LINE	FORMATTER / Option to control when a record declaration should be kept on one line - option id: "org.eclipse.jdt.core.formatter.keep_record_declaration_on_one_line" - possible values: { ONE_LINE_NEVER, ONE_LINE_IF_EMPTY, ONE_LINE_IF_SINGLE_ITEM, ONE_LINE_ALWAYS, ONE_LINE_PRESERVE } - default: ONE_LINE_NEVER
static final String	FORMATTER_KEEP_SIMPLE_DO_WHILE_BODY_ON_SAME_LINE	FORMATTER / Option to keep a simple 'do-while' loop body on the same line - option id: "org.eclipse.jdt.core.formatter.keep_simple_do_while_body_on_same_line" - possible values: { TRUE, FALSE } - default: FALSE
static final String	FORMATTER_KEEP_SIMPLE_FOR_BODY_ON_SAME_LINE	FORMATTER / Option to keep a simple 'for' loop body on the same line - option id: "org.eclipse.jdt.core.formatter.keep_simple_for_body_on_same_line" - possible values: { TRUE, FALSE } - default: FALSE
static final String	FORMATTER_KEEP_SIMPLE_GETTER_SETTER_ON_ONE_LINE	FORMATTER / Option to always keep simple getters and setters on one line, in addition to the #FORMATTER_KEEP_METHOD_BODY_ON_ONE_LINE option - option id: "org.eclipse.jdt.core.formatter.keep_simple_getter_setter_on_one_line" - possible values: { TRUE, FALSE } - default: FALSE
static final String	FORMATTER_KEEP_SIMPLE_IF_ON_ONE_LINE	FORMATTER / Option to keep simple if statement on the one line - option id: "org.eclipse.jdt.core.formatter.keep_imple_if_on_one_line" - possible values: { TRUE, FALSE } - default: FALSE
static final String	FORMATTER_KEEP_SIMPLE_WHILE_BODY_ON_SAME_LINE	FORMATTER / Option to keep a simple 'while' loop body on the same line - option id: "org.eclipse.jdt.core.formatter.keep_simple_while_body_on_same_line" - possible values: { TRUE, FALSE } - default: FALSE

static final String	FORMATTER_KEEP_SWITCH_BODY_BLOCK_ON_ONE_LINE	FORMATTER / Option to control when the body of a switch statement/ expression with arrows should be kept on one line - option id: "org.eclipse.jdt.core.formatter.keep_switch_body_block_on_one_line" - possible values: { ONE_LINE_NEVER, ONE_LINE_IF_EMPTY, ONE_LINE_IF_SINGLE_ITEM, ONE_LINE_ALWAYS, ONE_LINE_PRESERVE } - default: ONE_LINE_NEVER
static final String	FORMATTER_KEEP_SWITCH_CASE_WITH_ARROW_ON_ONE_LINE	FORMATTER / Option to control when a block following a switch case with arrow should be kept on one line - option id: "org.eclipse.jdt.core.formatter.keep_switch_case_with_arrow_on_one_line" - possible values: { ONE_LINE_NEVER, ONE_LINE_IF_EMPTY, ONE_LINE_IF_SINGLE_ITEM, ONE_LINE_ALWAYS, ONE_LINE_PRESERVE } - default: ONE_LINE_NEVER
static final String	FORMATTER_KEEP_THEN_STATEMENT_ON_SAME_LINE	FORMATTER / Option to keep then statement on the same line - option id: "org.eclipse.jdt.core.formatter.keep_then_statement_on_same_line" - possible values: { TRUE, FALSE } - default: FALSE
static final String	FORMATTER_KEEP_TYPE_DECLARATION_ON_ONE_LINE	FORMATTER / Option to control when a type declaration should be kept on one line - option id: "org.eclipse.jdt.core.formatter.keep_type_declaration_on_one_line" - possible values: { ONE_LINE_NEVER, ONE_LINE_IF_EMPTY, ONE_LINE_IF_SINGLE_ITEM, ONE_LINE_ALWAYS, ONE_LINE_PRESERVE } - default: ONE_LINE_NEVER
static final String	FORMATTER_LINE_SPLIT	FORMATTER / Option to specify the length of the page.
static final String	FORMATTER_NEVER_INDENT_BLOCK_COMMENTS_ON_FIRST_COLUMN	FORMATTER / Option to indent block comments that start on the first column - option id: "org.eclipse.jdt.core.formatter.formatter.never_indent_block_comments_on - possible values: { TRUE, FALSE } - default: FALSE
static final String	FORMATTER_NEVER_INDENT_LINE_COMMENTS_ON_FIRST_COLUMN	FORMATTER / Option to indent line comments that start on the first column - option id: "org.eclipse.jdt.core.formatter.formatter.never_indent_line_comments_on_ - possible values: { TRUE, FALSE } - default: FALSE
static final String	FORMATTER_NUMBER_OF_EMPTY_LINES_TO_PRESERVE	FORMATTER / Option to specify the number of empty lines to preserve - option id: "org.eclipse.jdt.core.formatter.number_of_empty_lines_to_preserve" - possible values: "<n>", where n is zero or a positive integer - default: "0"
static final String	FORMATTER_PARENTHESES_POSITIONS_IN_ANNOTATION	FORMATTER / Option to position parentheses in annotations - option id: "org.eclipse.jdt.core.formatter.parentheses_positions_in_annotation" - possible values: { COMMON_LINES, SEPARATE_LINES_IF_NOT_EMPTY, SEPARATE_LINES_IF_WRAPPED, SEPARATE_LINES, PRESERVE_POSITIONS } - default: COMMON_LINES
static final String	FORMATTER_PARENTHESES_POSITIONS_IN_CATCH_CLAUSE	FORMATTER / Option to position parentheses in catch clauses - option id: "org.eclipse.jdt.core.formatter.parentheses_positions_in_catch_clause" - possible values: { COMMON_LINES, SEPARATE_LINES_IF_WRAPPED, SEPARATE_LINES, PRESERVE_POSITIONS } - default: COMMON_LINES
static final String	FORMATTER_PARENTHESES_POSITIONS_IN_ENUM_CONSTANT_DECLARATION	FORMATTER / Option to position parentheses in enum constant declarations - option id: "org.eclipse.jdt.core.formatter.parentheses_positions_in_enum_constant_d - possible values: { COMMON_LINES, SEPARATE_LINES_IF_NOT_EMPTY, SEPARATE_LINES_IF_WRAPPED, SEPARATE_LINES, PRESERVE_POSITIONS } - default: COMMON_LINES
static final String	FORMATTER_PARENTHESES_POSITIONS_IN_FOR_STATEMENT	FORMATTER / Option to position parentheses in 'for' statements - option id: "org.eclipse.jdt.core.formatter.parentheses_positions_in_for_statment" - possible values: { COMMON_LINES, SEPARATE_LINES_IF_WRAPPED, SEPARATE_LINES, PRESERVE_POSITIONS } - default: COMMON_LINES
static final String	FORMATTER_PARENTHESES_POSITIONS_IN_IF_WHILE_STATEMENT	FORMATTER / Option to position parentheses in 'if' and 'while' statements - option id: "org.eclipse.jdt.core.formatter.parentheses_positions_in_if_while_statemen - possible values: { COMMON_LINES, SEPARATE_LINES_IF_WRAPPED, SEPARATE_LINES, PRESERVE_POSITIONS } - default: COMMON_LINES

static final String	FORMATTER_PARENTHESES_POSITIONS_IN_LAMBDA_DECLARATION	FORMATTER / Option to position parentheses in lambda declarations - option id: "org.eclipse.jdt.core.formatter.parentheses_positions_in_lambda_declaratio - possible values: { COMMON_LINES, SEPARATE_LINES_IF_NOT_EMPTY, SEPARATE_LINES_IF_WRAPPED, SEPARATE_LINES, PRESERVE_POSITIONS } - default: COMMON_LINES
static final String	FORMATTER_PARENTHESES_POSITIONS_IN_METHOD_DECLARATION	FORMATTER / Option to position parentheses in method declarations - option id: "org.eclipse.jdt.core.formatter.parentheses_positions_in_method_delcaratio - possible values: { COMMON_LINES, SEPARATE_LINES_IF_NOT_EMPTY, SEPARATE_LINES_IF_WRAPPED, SEPARATE_LINES, PRESERVE_POSITIONS } - default: COMMON_LINES
static final String	FORMATTER_PARENTHESES_POSITIONS_IN_METHOD_INVOCATION	FORMATTER / Option to position parentheses in method invocations - option id: "org.eclipse.jdt.core.formatter.parentheses_positions_in_method_invocatio - possible values: { COMMON_LINES, SEPARATE_LINES_IF_NOT_EMPTY, SEPARATE_LINES_IF_WRAPPED, SEPARATE_LINES, PRESERVE_POSITIONS } - default: COMMON_LINES
static final String	FORMATTER_PARENTHESES_POSITIONS_IN_RECORD_DECLARATION	FORMATTER / Option to position parentheses in record declarations - option id: "org.eclipse.jdt.core.formatter.parentheses_positions_in_record_declaratio - possible values: { COMMON_LINES, SEPARATE_LINES_IF_NOT_EMPTY, SEPARATE_LINES_IF_WRAPPED, SEPARATE_LINES, PRESERVE_POSITIONS } - default: COMMON_LINES
static final String	FORMATTER_PARENTHESES_POSITIONS_IN_SWITCH_STATEMENT	FORMATTER / Option to position parentheses in 'switch' statements - option id: "org.eclipse.jdt.core.formatter.parentheses_positions_in_switch_statement" - possible values: { COMMON_LINES, SEPARATE_LINES_IF_WRAPPED, SEPARATE_LINES, PRESERVE_POSITIONS } - default: COMMON_LINES
static final String	FORMATTER_PARENTHESES_POSITIONS_IN_TRY_CLAUSE	FORMATTER / Option to position parentheses in try clauses - option id: "org.eclipse.jdt.core.formatter.parentheses_positions_in_try_clause" - possible values: { COMMON_LINES, SEPARATE_LINES_IF_WRAPPED, SEPARATE_LINES, PRESERVE_POSITIONS } - default: COMMON_LINES
static final String	FORMATTER_PUT_EMPTY_STATEMENT_ON_NEW_LINE	FORMATTER / Option to specify whether or not empty statement should be on a new line - option id: "org.eclipse.jdt.core.formatter.put_empty_statement_on_new_line" - possible values: { TRUE, FALSE } - default: FALSE
static final String	FORMATTER_TAB_CHAR	FORMATTER / Option to specify the tabulation size - option id: "org.eclipse.jdt.core.formatter.tabulation.char" - possible values: { TAB, SPACE, MIXED } - default: TAB
static final String	FORMATTER_TAB_SIZE	FORMATTER / Option to specify the equivalent number of spaces that represents one tabulation - option id: "org.eclipse.jdt.core.formatter.tabulation.size" - possible values: "<n>", where n is zero or a positive integer - default: "4"
static final String	FORMATTER_TEXT_BLOCK_INDENTATION	FORMATTER / Option to specify how text blocks are indented - option id: "org.eclipse.jdt.core.formatter.text_block_indentation" - possible values: { INDENT_PRESERVE, INDENT_BY_ONE, INDENT_DEFAULT, INDENT_ON_COLUMN } - default: INDENT_DEFAULT
static final String	FORMATTER_USE_ON_OFF_TAGS	FORMATTER / Option to use the disabling and enabling tags defined respectively by the FORMATTER_DISABLING_TAG and the FORMATTER_ENABLING_TAG options
static final String	FORMATTER_USE_TABS_ONLY_FOR_LEADING_INDENTATIONS	FORMATTER / Option to use tabulations for indentation and spaces for line wrapping - option id: "org.eclipse.jdt.core.formatter.use_tabs_only_for_leading_indentations" - possible values: { TRUE, FALSE } - default: FALSE
static final String	FORMATTER_WRAP_BEFORE_ADDITIVE_OPERATOR	FORMATTER / Option to wrap before the additive operator (+, -) - option id: "org.eclipse.jdt.core.formatter.wrap_before_additive_operator" - possible values: { TRUE, FALSE } - default: TRUE
static final String	FORMATTER_WRAP_BEFORE_ASSERTION_MESSAGE_OPERATOR	FORMATTER / Option to wrap before the assertion message operator - option id: "org.eclipse.jdt.core.formatter.wrap_before_assertion_message_operator" - possible values: { TRUE, FALSE } - default: FALSE

static final String	FORMATTER_WRAP_BEFORE_ASSIGNMENT_OPERATOR	FORMATTER / Option to wrap before the assignment operator - option id: "org.eclipse.jdt.core.formatter.wrap_before_assignment_operator" - possible values: { TRUE, FALSE } - default: TRUE
static final String	FORMATTER_WRAP_BEFORE_BINARY_OPERATOR	**Deprecated.** Use the new options instead: FORMATTER_WRAP_BEFORE_MULTIPLICATIVE_OPERATOR, FORMATTER_WRAP_BEFORE_ADDITIVE_OPERATOR, FORMATTER_WRAP_BEFORE_STRING_CONCATENATION, FORMATTER_WRAP_BEFORE_BITWISE_OPERATOR, FORMATTER_WRAP_BEFORE_LOGICAL_OPERATOR
static final String	FORMATTER_WRAP_BEFORE_BITWISE_OPERATOR	FORMATTER / Option to wrap before the bitwise operator (&, ^, |) - option id: "org.eclipse.jdt.core.formatter.wrap_before_bitwise_operator" - possible values: { TRUE, FALSE } - default: TRUE
static final String	FORMATTER_WRAP_BEFORE_CONDITIONAL_OPERATOR	FORMATTER / Option to wrap before the '?'
static final String	FORMATTER_WRAP_BEFORE_LOGICAL_OPERATOR	FORMATTER / Option to wrap before the logical operator (&&, ||) - option id: "org.eclipse.jdt.core.formatter.wrap_before_logical_operator" - possible values: { TRUE, FALSE } - default: TRUE
static final String	FORMATTER_WRAP_BEFORE_MULTIPLICATIVE_OPERATOR	FORMATTER / Option to wrap before the multiplicative operator (*, /, %) - option id: "org.eclipse.jdt.core.formatter.wrap_before_multiplicative_operator" - possible values: { TRUE, FALSE } - default: TRUE
static final String	FORMATTER_WRAP_BEFORE_OR_OPERATOR_MULTICATCH	FORMATTER / Option to wrap before the '|' operator in multi-catch statements - option id: "org.eclipse.jdt.core.formatter.wrap_before_or_operator_multicatch" - possible values: { TRUE, FALSE } - default: TRUE
static final String	FORMATTER_WRAP_BEFORE_RELATIONAL_OPERATOR	FORMATTER / Option to wrap before the relational operator (<, >, <=, >=, ==, !=) - option id: "org.eclipse.jdt.core.formatter.wrap_before_" - possible values: { TRUE, FALSE } - default: TRUE
static final String	FORMATTER_WRAP_BEFORE_SHIFT_OPERATOR	FORMATTER / Option to wrap before the shift operator (<<, >>, >>>) - option id: "org.eclipse.jdt.core.formatter.wrap_before_shift_operator" - possible values: { TRUE, FALSE } - default: TRUE
static final String	FORMATTER_WRAP_BEFORE_STRING_CONCATENATION	FORMATTER / Option to wrap before the string concatenation operator - option id: "org.eclipse.jdt.core.formatter.wrap_before_string_concatenation" - possible values: { TRUE, FALSE } - default: TRUE
static final String	FORMATTER_WRAP_BEFORE_SWITCH_CASE_ARROW_OPERATOR	FORMATTER / Option to wrap before the arrow operator (->) in switch case - option id: "org.eclipse.jdt.core.formatter.wrap_before_switch_case_arrow_operator" - possible values: { TRUE, FALSE } - default: FALSE
static final String	FORMATTER_WRAP_OUTER_EXPRESSIONS_WHEN_NESTED	FORMATTER / Option to wrap outer expressions in nested expressions - option id: "org.eclipse.jdt.core.formatter.wrap_outer_expressions_when_nested" - possible values: { TRUE, FALSE } - default: TRUE
static final int	INDENT_BY_ONE	FORMATTER / The wrapping is done by indenting by one compare to the current indentation.
static final int	INDENT_DEFAULT	FORMATTER / The wrapping is done by using the current indentation.
static final int	INDENT_ON_COLUMN	FORMATTER / The wrapping is done by indenting on column under the splitting location.
static final int	INDENT_PRESERVE	FORMATTER / Indentation is not touched, it's preserved from original source.
static final String	MIXED	FORMATTER / Possible value for the option FORMATTER_TAB_CHAR
static final String	NEXT_LINE	FORMATTER / Value to set a brace location at the start of the next line with the right indentation.
static final String	NEXT_LINE_ON_WRAP	FORMATTER / Value to set a brace location at the start of the next line if a wrapping occured.
static final String	NEXT_LINE_SHIFTED	FORMATTER / Value to set a brace location at the start of the next line with an extra indentation.
static final String	ONE_LINE_ALWAYS	FORMATTER / Value to always keep braced code on one line, as long as it doesn't exceed the line width limit.
static final String	ONE_LINE_IF_EMPTY	FORMATTER / Value to keep braced code on one line only if it's empty.
static final String	ONE_LINE_IF_SINGLE_ITEM	FORMATTER / Value to keep braced code on one line if it contains at most a single item.

static final String	ONE_LINE_NEVER	FORMATTER / Value to never keep braced code on one line.
static final String	ONE_LINE_PRESERVE	FORMATTER / Value to keep braced code on one line as long as it doesn't exceed the line width limit and it was already in one line in the original source.
static final String	PRESERVE_POSITIONS	FORMATTER / Value to set opening and closing parentheses location to be preserved from the original source.
static final String	SEPARATE_LINES	FORMATTER / Value to set parentheses location on separate lines from their contents, that is put a line break after the opening parenthesis and before the closing parenthesis.
static final String	SEPARATE_LINES_IF_NOT_EMPTY	FORMATTER / Value to set opening and closing parentheses location on a common line if the parentheses are empty and otherwise in separate lines from their contents.
static final String	SEPARATE_LINES_IF_WRAPPED	FORMATTER / Value to set opening and closing parentheses location on separate lines from their contents if the contents are wrapped, and in common line if they fit in line width.
static final String	TRUE	FORMATTER / Value to set an option to true.
static final int	WRAP_COMPACT	FORMATTER / The wrapping is done using as few lines as possible.
static final int	WRAP_COMPACT_FIRST_BREAK	FORMATTER / The wrapping is done putting the first element on a new line and then wrapping next elements using as few lines as possible.
static final int	WRAP_NEXT_PER_LINE	FORMATTER / The wrapping is done by putting each element on its own line except the first element.
static final int	WRAP_NEXT_SHIFTED	FORMATTER / The wrapping is done by putting each element on its own line.
static final int	WRAP_NO_SPLIT	FORMATTER / Value to disable alignment.
static final int	WRAP_ONE_PER_LINE	FORMATTER / The wrapping is done by putting each element on its own line.

Constructor Summary

Constructors

Constructor	Description
DefaultCodeFormatterConstants()	

Method Summary

All Methods **Static Methods** **Concrete Methods**

Modifier and Type	Method	Description
static String	createAlignmentValue(boolean forceSplit, int wrapStyle)	Create a new alignment value according to the given values.
static String	createAlignmentValue(boolean forceSplit, int wrapStyle, int indentStyle)	Create a new alignment value according to the given values.
static Map <String ,String >	getEclipse21Settings()	Returns the formatter settings that most closely approximate the default formatter settings of Eclipse version 2.1.
static Map <String ,String >	getEclipseDefaultSettings()	Returns the default Eclipse formatter settings
static boolean	getForceWrapping(String value)	Return the force value of the given alignment value.
static int	getIndentStyle(String value)	Return the indentation style of the given alignment value.
static Map <String ,String >	getJavaConventionsSettings()	Returns the settings according to the Java conventions.
static int	getWrappingStyle(String value)	Return the wrapping style of the given alignment value.
static String	setForceWrapping(String value, boolean force)	Set the force value of the given alignment value and return the new value.
static String	setIndentStyle(String value, int indentStyle)	Set the indentation style of the given alignment value and return the new value.
static String	setWrappingStyle(String value, int wrappingStyle)	Set the wrapping style of the given alignment value and return the new value.

Methods inherited from class java.lang.Object

clone , equals , finalize , getClass , hashCode , notify , notifyAll , toString , wait , wait , wait

Field Details

END_OF_LINE

```
public static final String  END_OF_LINE
```

FORMATTER / Value to set a brace location at the end of a line.

Since:

3.0

See Also:

FORMATTER_BRACE_POSITION_FOR_ANONYMOUS_TYPE_DECLARATION,
FORMATTER_BRACE_POSITION_FOR_ARRAY_INITIALIZER,
FORMATTER_BRACE_POSITION_FOR_BLOCK,
FORMATTER_BRACE_POSITION_FOR_CONSTRUCTOR_DECLARATION,
FORMATTER_BRACE_POSITION_FOR_METHOD_DECLARATION,
FORMATTER_BRACE_POSITION_FOR_RECORD_CONSTRUCTOR,
FORMATTER_BRACE_POSITION_FOR_RECORD_DECLARATION,
FORMATTER_BRACE_POSITION_FOR_SWITCH,
FORMATTER_BRACE_POSITION_FOR_TYPE_DECLARATION,
FORMATTER_BRACE_POSITION_FOR_LAMBDA_BODY,
Constant Field Values

FALSE

```
public static final String  FALSE
```

FORMATTER / Value to set an option to false.

Since:

3.0

See Also:

Constant Field Values

FORMATTER_ALIGN_TYPE_MEMBERS_ON_COLUMNS

```
public static final String  FORMATTER_ALIGN_TYPE_MEMBERS_ON_COLUMNS
```

FORMATTER / Option to align type members of a type declaration on column

```
     - option id:         "org.eclipse.jdt.core.formatter.align_type_members_on_columns"
     - possible values:   { TRUE, FALSE }
     - default:           FALSE
```

Since:

3.0

See Also:

TRUE, FALSE, Constant Field Values

FORMATTER_ALIGN_VARIABLE_DECLARATIONS_ON_COLUMNS

```
public static final String  FORMATTER_ALIGN_VARIABLE_DECLARATIONS_ON_COLUMNS
```

```
 FORMATTER / Option to align variable declarations on column
     - option id:         "org.eclipse.jdt.core.formatter.align_variable_declarations_on_columns"
     - possible values:   { TRUE, FALSE }
     - default:           FALSE
```

Since:

3.15

See Also:

TRUE, FALSE, Constant Field Values

FORMATTER_ALIGN_ASSIGNMENT_STATEMENTS_ON_COLUMNS

```
public static final String  FORMATTER_ALIGN_ASSIGNMENT_STATEMENTS_ON_COLUMNS
```

```
 FORMATTER / Option to align assignment statements on column
     - option id:         "org.eclipse.jdt.core.formatter.align_assignment_statements_on_columns"
     - possible values:   { TRUE, FALSE }
     - default:           FALSE
```

Since:

3.15

See Also:

TRUE, FALSE, Constant Field Values

FORMATTER_ALIGN_ARROWS_IN_SWITCH_ON_COLUMNS

```
public static final String  FORMATTER_ALIGN_ARROWS_IN_SWITCH_ON_COLUMNS
```

```
 FORMATTER / Option to align arrows in switch on column
     - option id:         "org.eclipse.jdt.core.formatter.align_arrows_in_switch_on_columns"
     - possible values:   { TRUE, FALSE }
     - default:           FALSE
```

Since:

3.37

See Also:

TRUE, FALSE, Constant Field Values

FORMATTER_ALIGN_WITH_SPACES

```
public static final String  FORMATTER_ALIGN_WITH_SPACES
```

```
 FORMATTER / Option to use spaces when aligning members, independent of selected tabulation character
     - option id:         "org.eclipse.jdt.core.formatter.align_with_spaces"
     - possible values:   { TRUE, FALSE }
     - default:           FALSE
```

Since:

3.15

See Also:

TRUE, FALSE, Constant Field Values

FORMATTER_ALIGN_FIELDS_GROUPING_BLANK_LINES

```
public static final String  FORMATTER_ALIGN_FIELDS_GROUPING_BLANK_LINES
```

```
 FORMATTER / Option to affect aligning on columns: groups of items are aligned independently
```

```
 if they are separated by at least the selected number of blank lines.
 Note: since 3.15 the 'fields' part is a (potentially misleading) residue as this option
 affects other types of aligning on columns as well.
     - option id:         "org.eclipse.jdt.core.formatter.align_fields_grouping_blank_lines"
     - possible values:   "<n>", where n is a positive integer
     - default:           Integer.MAX_VALUE
```

Since:

3.12

See Also:

Constant Field Values

FORMATTER_ALIGN_SELECTOR_IN_METHOD_INVOCATION_ON_EXPRESSION_FIRST_LINE

```
public static final String  FORMATTER_ALIGN_SELECTOR_IN_METHOD_INVOCATION_ON_EXPRESSION_FIRST_LINE
```

```
 FORMATTER / Option to indent method invocation chains based on the first line of the base
 expression rather than the last line.
     - option id:         "org.eclipse.jdt.core.formatter.align_selector_in_method_invocation_on_expression_first_line"
     - possible values:   { TRUE, FALSE }
     - default:           TRUE
```

Since:

3.29

See Also:

`TRUE`, `FALSE`, Constant Field Values

FORMATTER_ALIGNMENT_FOR_ANNOTATIONS_ON_ENUM_CONSTANT

```
public static final String  FORMATTER_ALIGNMENT_FOR_ANNOTATIONS_ON_ENUM_CONSTANT
```

```
 FORMATTER / Option for alignment of annotations on enum constant declaration
     - option id:         "org.eclipse.jdt.core.formatter.alignment_for_annotations_on_enum_constant"
     - possible values:   values returned by createAlignmentValue(boolean, int) call
     - default:           createAlignmentValue(true, WRAP_ONE_PER_LINE)
```

Since:

3.24

See Also:

`createAlignmentValue(boolean, int)`,
Constant Field Values

FORMATTER_ALIGNMENT_FOR_ANNOTATIONS_ON_FIELD

```
public static final String  FORMATTER_ALIGNMENT_FOR_ANNOTATIONS_ON_FIELD
```

```
 FORMATTER / Option for alignment of annotations on field declaration
     - option id:         "org.eclipse.jdt.core.formatter.alignment_for_annotations_on_field"
     - possible values:   values returned by createAlignmentValue(boolean, int) call
     - default:           createAlignmentValue(true, WRAP_ONE_PER_LINE)
```

Since:

3.24

See Also:

`createAlignmentValue(boolean, int)`,
Constant Field Values

FORMATTER_ALIGNMENT_FOR_ANNOTATIONS_ON_METHOD

```
public static final String  FORMATTER_ALIGNMENT_FOR_ANNOTATIONS_ON_METHOD
```

```
 FORMATTER / Option for alignment of annotations on method declaration
     - option id:         "org.eclipse.jdt.core.formatter.alignment_for_annotations_on_method"
     - possible values:   values returned by createAlignmentValue(boolean, int) call
     - default:           createAlignmentValue(true, WRAP_ONE_PER_LINE)
```

Since:

3.24

See Also:

createAlignmentValue(boolean, int),
Constant Field Values

FORMATTER_ALIGNMENT_FOR_ANNOTATIONS_ON_PACKAGE

```
public static final String  FORMATTER_ALIGNMENT_FOR_ANNOTATIONS_ON_PACKAGE
```

```
 FORMATTER / Option for alignment of annotations on package declaration
     - option id:         "org.eclipse.jdt.core.formatter.alignment_for_annotations_on_package"
     - possible values:   values returned by createAlignmentValue(boolean, int) call
     - default:           createAlignmentValue(true, WRAP_ONE_PER_LINE)
```

Since:

3.24

See Also:

createAlignmentValue(boolean, int),
Constant Field Values

FORMATTER_ALIGNMENT_FOR_ANNOTATIONS_ON_TYPE

```
public static final String  FORMATTER_ALIGNMENT_FOR_ANNOTATIONS_ON_TYPE
```

```
 FORMATTER / Option for alignment of annotations on type declaration
     - option id:         "org.eclipse.jdt.core.formatter.alignment_for_annotations_on_type"
     - possible values:   values returned by createAlignmentValue(boolean, int) call
     - default:           createAlignmentValue(true, WRAP_ONE_PER_LINE)
```

Since:

3.24

See Also:

createAlignmentValue(boolean, int),
Constant Field Values

FORMATTER_ALIGNMENT_FOR_TYPE_ANNOTATIONS

```
public static final String  FORMATTER_ALIGNMENT_FOR_TYPE_ANNOTATIONS
```

```
 FORMATTER / Option for alignment of type annotations
     - option id:         "org.eclipse.jdt.core.formatter.alignment_for_type_annotations"
     - possible values:   values returned by createAlignmentValue(boolean, int) call
     - default:           createAlignmentValue(false, WRAP_NO_SPLIT)
```

Since:

3.24

See Also:

createAlignmentValue(boolean, int),
Constant Field Values

FORMATTER_ALIGNMENT_FOR_ANNOTATIONS_ON_PARAMETER

```
public static final String  FORMATTER_ALIGNMENT_FOR_ANNOTATIONS_ON_PARAMETER
```

```
 FORMATTER / Option for alignment of annotations on parameter
     - option id:         "org.eclipse.jdt.core.formatter.alignment_for_annotations_on_parameter"
     - possible values:   values returned by createAlignmentValue(boolean, int) call
     - default:           createAlignmentValue(false, WRAP_NO_SPLIT)
```

Since:

3.24

See Also:

createAlignmentValue(boolean, int),
Constant Field Values

FORMATTER_ALIGNMENT_FOR_ANNOTATIONS_ON_LOCAL_VARIABLE

```
public static final String  FORMATTER_ALIGNMENT_FOR_ANNOTATIONS_ON_LOCAL_VARIABLE
```

```
 FORMATTER / Option for alignment of annotations on local variable
     - option id:         "org.eclipse.jdt.core.formatter.alignment_for_annotations_on_local_variable"
     - possible values:   values returned by createAlignmentValue(boolean, int) call
     - default:           createAlignmentValue(true, WRAP_ONE_PER_LINE)
```

Since:

3.24

See Also:

createAlignmentValue(boolean, int),
Constant Field Values

FORMATTER_ALIGNMENT_FOR_ARGUMENTS_IN_ALLOCATION_EXPRESSION

```
public static final String  FORMATTER_ALIGNMENT_FOR_ARGUMENTS_IN_ALLOCATION_EXPRESSION
```

```
 FORMATTER / Option for alignment of arguments in allocation expression
     - option id:         "org.eclipse.jdt.core.formatter.alignment_for_arguments_in_allocation_expression"
     - possible values:   values returned by createAlignmentValue(boolean, int, int) call
     - default:           createAlignmentValue(false, WRAP_COMPACT, INDENT_DEFAULT)
```

Since:

3.0

See Also:

createAlignmentValue(boolean, int, int),
Constant Field Values

FORMATTER_ALIGNMENT_FOR_ARGUMENTS_IN_ENUM_CONSTANT

```
public static final String  FORMATTER_ALIGNMENT_FOR_ARGUMENTS_IN_ENUM_CONSTANT
```

```
 FORMATTER / Option for alignment of arguments in enum constant
     - option id:         "org.eclipse.jdt.core.formatter.alignment_for_arguments_in_enum_constant"
     - possible values:   values returned by createAlignmentValue(boolean, int, int) call
     - default:           createAlignmentValue(false, WRAP_COMPACT, INDENT_DEFAULT)
```

Since:

3.1

See Also:

createAlignmentValue(boolean, int, int),
Constant Field Values

FORMATTER_ALIGNMENT_FOR_ARGUMENTS_IN_ANNOTATION

```
public static final String  FORMATTER_ALIGNMENT_FOR_ARGUMENTS_IN_ANNOTATION
```

```
 FORMATTER / Option for alignment of arguments in annotation
     - option id:         "org.eclipse.jdt.core.formatter.alignment_for_arguments_in_annotation"
     - possible values:   values returned by createAlignmentValue(boolean, int, int) call
     - default:           createAlignmentValue(false, WRAP_NO_SPLIT, INDENT_DEFAULT)
```

Since:

3.6

See Also:

createAlignmentValue(boolean, int, int),
Constant Field Values

FORMATTER_ALIGNMENT_FOR_ARGUMENTS_IN_EXPLICIT_CONSTRUCTOR_CALL

```
public static final String  FORMATTER_ALIGNMENT_FOR_ARGUMENTS_IN_EXPLICIT_CONSTRUCTOR_CALL
```

```
 FORMATTER / Option for alignment of arguments in explicit constructor call
     - option id:         "org.eclipse.jdt.core.formatter.alignment_for_arguments_in_explicit_constructor_call"
     - possible values:   values returned by createAlignmentValue(boolean, int, int) call
```

```
     - default:          createAlignmentValue(false, WRAP_COMPACT, INDENT_DEFAULT)
```

Since:

3.0

See Also:

createAlignmentValue(boolean, int, int),
Constant Field Values

FORMATTER_ALIGNMENT_FOR_ARGUMENTS_IN_METHOD_INVOCATION

```
public static final String  FORMATTER_ALIGNMENT_FOR_ARGUMENTS_IN_METHOD_INVOCATION

 FORMATTER / Option for alignment of arguments in method invocation
     - option id:         "org.eclipse.jdt.core.formatter.alignment_for_arguments_in_method_invocation"
     - possible values:   values returned by createAlignmentValue(boolean, int, int) call
     - default:          createAlignmentValue(false, WRAP_COMPACT, INDENT_DEFAULT)
```

Since:

3.0

See Also:

createAlignmentValue(boolean, int, int),
Constant Field Values

FORMATTER_ALIGNMENT_FOR_ARGUMENTS_IN_QUALIFIED_ALLOCATION_EXPRESSION

```
public static final String  FORMATTER_ALIGNMENT_FOR_ARGUMENTS_IN_QUALIFIED_ALLOCATION_EXPRESSION

 FORMATTER / Option for alignment of arguments in qualified allocation expression
     - option id:         "org.eclipse.jdt.core.formatter.alignment_for_arguments_in_qualified_allocation_expression"
     - possible values:   values returned by createAlignmentValue(boolean, int, int) call
     - default:          createAlignmentValue(false, WRAP_COMPACT, INDENT_DEFAULT)
```

Since:

3.0

See Also:

createAlignmentValue(boolean, int, int),
Constant Field Values

FORMATTER_ALIGNMENT_FOR_ASSIGNMENT

```
public static final String  FORMATTER_ALIGNMENT_FOR_ASSIGNMENT

 FORMATTER / Option for alignment of assignment (=, +=, -=, *=, /=, %=, &=, ^=, |=, <<=, >>=, >>>=)
     - option id:         "org.eclipse.jdt.core.formatter.alignment_for_assignment"
     - possible values:   values returned by createAlignmentValue(boolean, int, int) call
     - default:          createAlignmentValue(false, WRAP_NO_SPLIT, INDENT_DEFAULT)
```

Since:

3.2

See Also:

createAlignmentValue(boolean, int, int),
Constant Field Values

FORMATTER_ALIGNMENT_FOR_ASSERTION_MESSAGE

```
public static final String  FORMATTER_ALIGNMENT_FOR_ASSERTION_MESSAGE

 FORMATTER / Option for alignment of assertion message separator (:)
     - option id:         "org.eclipse.jdt.core.formatter.alignment_for_assertion_message"
     - possible values:   values returned by createAlignmentValue(boolean, int, int) call
     - default:          createAlignmentValue(false, WRAP_NO_SPLIT, INDENT_DEFAULT)
```

Since:

3.23

See Also:

createAlignmentValue(boolean, int, int),
Constant Field Values

FORMATTER_ALIGNMENT_FOR_MULTIPLICATIVE_OPERATOR

```
public static final String  FORMATTER_ALIGNMENT_FOR_MULTIPLICATIVE_OPERATOR
```

```
 FORMATTER / Option for alignment of expressions with multiplicative operators (*, /, %)
     - option id:         "org.eclipse.jdt.core.formatter.alignment_for_multiplicative_operator"
     - possible values:   values returned by createAlignmentValue(boolean, int, int) call
     - default:           createAlignmentValue(false, WRAP_COMPACT, INDENT_DEFAULT)
```

Since:

3.17

See Also:

createAlignmentValue(boolean, int, int),
Constant Field Values

FORMATTER_ALIGNMENT_FOR_ADDITIVE_OPERATOR

```
public static final String  FORMATTER_ALIGNMENT_FOR_ADDITIVE_OPERATOR
```

```
 FORMATTER / Option for alignment of expressions with additive operators (+, -)
     - option id:         "org.eclipse.jdt.core.formatter.alignment_for_additive_operator"
     - possible values:   values returned by createAlignmentValue(boolean, int, int) call
     - default:           createAlignmentValue(false, WRAP_COMPACT, INDENT_DEFAULT)
```

Since:

3.17

See Also:

createAlignmentValue(boolean, int, int),
Constant Field Values

FORMATTER_ALIGNMENT_FOR_STRING_CONCATENATION

```
public static final String  FORMATTER_ALIGNMENT_FOR_STRING_CONCATENATION
```

```
 FORMATTER / Option for alignment of string concatenation expressions
     - option id:         "org.eclipse.jdt.core.formatter.alignment_for_string_concatenation"
     - possible values:   values returned by createAlignmentValue(boolean, int, int) call
     - default:           createAlignmentValue(false, WRAP_COMPACT, INDENT_DEFAULT)
```

Since:

3.17

See Also:

createAlignmentValue(boolean, int, int),
Constant Field Values

FORMATTER_ALIGNMENT_FOR_SHIFT_OPERATOR

```
public static final String  FORMATTER_ALIGNMENT_FOR_SHIFT_OPERATOR
```

```
 FORMATTER / Option for alignment of expressions with shift operators (<<, >>, >>>)
     - option id:         "org.eclipse.jdt.core.formatter.alignment_for_shift_operator"
     - possible values:   values returned by createAlignmentValue(boolean, int, int) call
     - default:           createAlignmentValue(false, WRAP_COMPACT, INDENT_DEFAULT)
```

Since:

3.17

See Also:

createAlignmentValue(boolean, int, int),
Constant Field Values

FORMATTER_ALIGNMENT_FOR_RELATIONAL_OPERATOR

```
public static final String  FORMATTER_ALIGNMENT_FOR_RELATIONAL_OPERATOR
```

```
 FORMATTER / Option for alignment of expressions with relational operators (<, >, <=, >=, ==, !=)
     - option id:         "org.eclipse.jdt.core.formatter.alignment_for_relational_operator"
     - possible values:   values returned by createAlignmentValue(boolean, int, int) call
     - default:           createAlignmentValue(false, WRAP_COMPACT, INDENT_DEFAULT)
```

Since:

3.17

See Also:

createAlignmentValue(boolean, int, int),

Constant Field Values

FORMATTER_ALIGNMENT_FOR_BITWISE_OPERATOR

```
public static final String  FORMATTER_ALIGNMENT_FOR_BITWISE_OPERATOR
```

```
 FORMATTER / Option for alignment of expressions with bitwise operators (&, ^, |)
     - option id:         "org.eclipse.jdt.core.formatter.alignment_for_bitwise_operator"
     - possible values:   values returned by createAlignmentValue(boolean, int, int) call
     - default:           createAlignmentValue(false, WRAP_COMPACT, INDENT_DEFAULT)
```

Since:

3.17

See Also:

createAlignmentValue(boolean, int, int),
Constant Field Values

FORMATTER_ALIGNMENT_FOR_LOGICAL_OPERATOR

```
public static final String  FORMATTER_ALIGNMENT_FOR_LOGICAL_OPERATOR
```

```
 FORMATTER / Option for alignment of expressions with logical operators (&&, ||)
     - option id:         "org.eclipse.jdt.core.formatter.alignment_for_logical_operator"
     - possible values:   values returned by createAlignmentValue(boolean, int, int) call
     - default:           createAlignmentValue(false, WRAP_COMPACT, INDENT_DEFAULT)
```

Since:

3.17

See Also:

createAlignmentValue(boolean, int, int),
Constant Field Values

FORMATTER_ALIGNMENT_FOR_BINARY_EXPRESSION

```
public static final String  FORMATTER_ALIGNMENT_FOR_BINARY_EXPRESSION
```

Deprecated.
Use new settings instead: FORMATTER_ALIGNMENT_FOR_MULTIPLICATIVE_OPERATOR, FORMATTER_ALIGNMENT_FOR_ADDITIVE_OPERATOR, FORMATTER_ALIGNMENT_FOR_STRING_CONCATENATION, FORMATTER_ALIGNMENT_FOR_BITWISE_OPERATOR, FORMATTER_ALIGNMENT_FOR_LOGICAL_OPERATOR

```
 FORMATTER / Option for alignment of binary expression
     - option id:         "org.eclipse.jdt.core.formatter.alignment_for_binary_expression"
     - possible values:   values returned by createAlignmentValue(boolean, int, int) call
     - default:           createAlignmentValue(false, WRAP_COMPACT, INDENT_DEFAULT)
```

Since:

3.0

See Also:

createAlignmentValue(boolean, int, int),
Constant Field Values

FORMATTER_ALIGNMENT_FOR_COMPACT_IF

```
public static final String  FORMATTER_ALIGNMENT_FOR_COMPACT_IF
```

```
 FORMATTER / Option for alignment of compact if
     - option id:         "org.eclipse.jdt.core.formatter.alignment_for_compact_if"
     - possible values:   values returned by createAlignmentValue(boolean, int, int) call
     - default:           createAlignmentValue(false, WRAP_ONE_PER_LINE, INDENT_BY_ONE)
```

Since:

3.0

See Also:

createAlignmentValue(boolean, int, int),
Constant Field Values

FORMATTER_ALIGNMENT_FOR_COMPACT_LOOP

```
public static final String  FORMATTER_ALIGNMENT_FOR_COMPACT_LOOP
```

```
 FORMATTER / Option for alignment of compact loops
     - option id:         "org.eclipse.jdt.core.formatter.alignment_for_compact_loops"
     - possible values:   values returned by createAlignmentValue(boolean, int, int) call
     - default:           createAlignmentValue(false, WRAP_ONE_PER_LINE, INDENT_BY_ONE)
```

Since:

3.15

See Also:

createAlignmentValue(boolean, int, int),
Constant Field Values

FORMATTER_ALIGNMENT_FOR_CONDITIONAL_EXPRESSION

```
public static final String  FORMATTER_ALIGNMENT_FOR_CONDITIONAL_EXPRESSION
```

```
 FORMATTER / Option for alignment of conditional expression
     - option id:         "org.eclipse.jdt.core.formatter.alignment_for_conditional_expression"
     - possible values:   values returned by createAlignmentValue(boolean, int, int) call
     - default:           createAlignmentValue(false, WRAP_ONE_PER_LINE, INDENT_DEFAULT)
```

Since:

3.0

See Also:

createAlignmentValue(boolean, int, int),
FORMATTER_ALIGNMENT_FOR_CONDITIONAL_EXPRESSION_CHAIN,
Constant Field Values

FORMATTER_ALIGNMENT_FOR_CONDITIONAL_EXPRESSION_CHAIN

```
public static final String  FORMATTER_ALIGNMENT_FOR_CONDITIONAL_EXPRESSION_CHAIN
```

```
 FORMATTER / Option for alignment of conditional expression chains. If disabled, chains are not recognized
             and only FORMATTER_ALIGNMENT_FOR_CONDITIONAL_EXPRESSION policy is used instead.
     - option id:         "org.eclipse.jdt.core.formatter.alignment_for_conditional_expression_chain"
     - possible values:   values returned by createAlignmentValue(boolean, int, int) call
     - default:           createAlignmentValue(false, WRAP_NO_SPLIT, INDENT_DEFAULT)
```

Since:

3.18

See Also:

createAlignmentValue(boolean, int, int),
FORMATTER_ALIGNMENT_FOR_CONDITIONAL_EXPRESSION,
Constant Field Values

FORMATTER_ALIGNMENT_FOR_ENUM_CONSTANTS

```
public static final String  FORMATTER_ALIGNMENT_FOR_ENUM_CONSTANTS
```

```
 FORMATTER / Option for alignment of enum constants
     - option id:        "org.eclipse.jdt.core.formatter.alignment_for_enum_constants"
     - possible values:  values returned by createAlignmentValue(boolean, int, int) call
     - default:          createAlignmentValue(false, WRAP_NO_SPLIT, INDENT_DEFAULT)
```

Since:

3.1

See Also:

createAlignmentValue(boolean, int, int),
Constant Field Values

FORMATTER_ALIGNMENT_FOR_EXPRESSIONS_IN_ARRAY_INITIALIZER

```
public static final String  FORMATTER_ALIGNMENT_FOR_EXPRESSIONS_IN_ARRAY_INITIALIZER
```

```
 FORMATTER / Option for alignment of expressions in array initializer
     - option id:         "org.eclipse.jdt.core.formatter.alignment_for_expressions_in_array_initializer"
     - possible values:   values returned by createAlignmentValue(boolean, int, int) call
     - default:           createAlignmentValue(false, WRAP_COMPACT, INDENT_DEFAULT)
```

Since:

3.0

See Also:

createAlignmentValue(boolean, int, int),
Constant Field Values

FORMATTER_ALIGNMENT_FOR_EXPRESSIONS_IN_FOR_LOOP_HEADER

```
public static final String  FORMATTER_ALIGNMENT_FOR_EXPRESSIONS_IN_FOR_LOOP_HEADER
```

```
 FORMATTER / Option for alignment of initialization, termination, and increment expressions in 'for'
             loop header
     - option id:         "org.eclipse.jdt.core.formatter.alignment_for_for_loop_header"
     - possible values:   values returned by createAlignmentValue(boolean, int, int) call
     - default:           createAlignmentValue(false, WRAP_NO_SPLIT, INDENT_DEFAULT)
```

Since:

3.12

See Also:

createAlignmentValue(boolean, int, int),
Constant Field Values

FORMATTER_ALIGNMENT_FOR_EXPRESSIONS_IN_SWITCH_CASE_WITH_ARROW

```
public static final String  FORMATTER_ALIGNMENT_FOR_EXPRESSIONS_IN_SWITCH_CASE_WITH_ARROW
```

```
 FORMATTER / Option for alignment of expressions in switch case with arrow
     - option id:         "org.eclipse.jdt.core.formatter.alignment_for_expressions_in_switch_case_with_arrow"
     - possible values:   values returned by createAlignmentValue(boolean, int, int) call
     - default:           createAlignmentValue(false, WRAP_COMPACT, INDENT_DEFAULT)
```

Since:

3.29

See Also:

createAlignmentValue(boolean, int, int),
Constant Field Values

FORMATTER_ALIGNMENT_FOR_EXPRESSIONS_IN_SWITCH_CASE_WITH_COLON

```
public static final String  FORMATTER_ALIGNMENT_FOR_EXPRESSIONS_IN_SWITCH_CASE_WITH_COLON
```

```
 FORMATTER / Option for alignment of expressions in switch case with colon
     - option id:         "org.eclipse.jdt.core.formatter.alignment_for_expressions_in_switch_case_with_colon"
     - possible values:   values returned by createAlignmentValue(boolean, int, int) call
     - default:           createAlignmentValue(false, WRAP_COMPACT, INDENT_DEFAULT)
```

Since:

3.29

See Also:

createAlignmentValue(boolean, int, int),
Constant Field Values

FORMATTER_ALIGNMENT_FOR_METHOD_DECLARATION

```
public static final String  FORMATTER_ALIGNMENT_FOR_METHOD_DECLARATION
```

```
 FORMATTER / Option for alignment of method declaration
     - option id:         "org.eclipse.jdt.core.formatter.alignment_for_method_declaration"
     - possible values:   values returned by createAlignmentValue(boolean, int, int) call
     - default:           createAlignmentValue(false, WRAP_NO_SPLIT, INDENT_DEFAULT)
```

Since:

3.6

See Also:

createAlignmentValue(boolean, int, int),
Constant Field Values

FORMATTER_ALIGNMENT_FOR_MODULE_STATEMENTS

```
public static final String  FORMATTER_ALIGNMENT_FOR_MODULE_STATEMENTS
```

```
 FORMATTER / Option for alignment of module statements
     - option id:         "org.eclipse.jdt.core.formatter.alignment_for_module_statements"
     - possible values:   values returned by createAlignmentValue(boolean, int, int) call
```

```
     - default:          createAlignmentValue(false, WRAP_COMPACT, INDENT_DEFAULT)
```

Since:

3.14

See Also:

createAlignmentValue(boolean, int, int),
Constant Field Values

FORMATTER_ALIGNMENT_FOR_MULTIPLE_FIELDS

```
public static final String  FORMATTER_ALIGNMENT_FOR_MULTIPLE_FIELDS

 FORMATTER / Option for alignment of multiple fields
     - option id:         "org.eclipse.jdt.core.formatter.alignment_for_multiple_fields"
     - possible values:   values returned by createAlignmentValue(boolean, int, int) call
     - default:          createAlignmentValue(false, WRAP_COMPACT, INDENT_DEFAULT)
```

Since:

3.0

See Also:

createAlignmentValue(boolean, int, int),
Constant Field Values

FORMATTER_ALIGNMENT_FOR_PARAMETERIZED_TYPE_REFERENCES

```
public static final String  FORMATTER_ALIGNMENT_FOR_PARAMETERIZED_TYPE_REFERENCES

 FORMATTER / Option for alignment of type arguments in parameterized type references
     - option id:         "org.eclipse.jdt.core.formatter.alignment_for_parameterized_type_references"
     - possible values:   values returned by createAlignmentValue(boolean, int, int) call
     - default:          createAlignmentValue(false, WRAP_NO_SPLIT, INDENT_DEFAULT)
```

Since:

3.12

See Also:

createAlignmentValue(boolean, int, int),
Constant Field Values

FORMATTER_ALIGNMENT_FOR_PARAMETERS_IN_CONSTRUCTOR_DECLARATION

```
public static final String  FORMATTER_ALIGNMENT_FOR_PARAMETERS_IN_CONSTRUCTOR_DECLARATION

 FORMATTER / Option for alignment of parameters in constructor declaration
     - option id:         "org.eclipse.jdt.core.formatter.alignment_for_parameters_in_constructor_declaration"
     - possible values:   values returned by createAlignmentValue(boolean, int, int) call
     - default:          createAlignmentValue(false, WRAP_COMPACT, INDENT_DEFAULT)
```

Since:

3.0

See Also:

createAlignmentValue(boolean, int, int),
Constant Field Values

FORMATTER_ALIGNMENT_FOR_PARAMETERS_IN_METHOD_DECLARATION

```
public static final String  FORMATTER_ALIGNMENT_FOR_PARAMETERS_IN_METHOD_DECLARATION

 FORMATTER / Option for alignment of parameters in method declaration
     - option id:         "org.eclipse.jdt.core.formatter.alignment_for_parameters_in_method_declaration"
     - possible values:   values returned by createAlignmentValue(boolean, int, int) call
     - default:          createAlignmentValue(false, WRAP_COMPACT, INDENT_DEFAULT)
```

Since:

3.0

See Also:

createAlignmentValue(boolean, int, int),
Constant Field Values

FORMATTER_ALIGNMENT_FOR_PERMITTED_TYPES_IN_TYPE_DECLARATION

```
public static final String  FORMATTER_ALIGNMENT_FOR_PERMITTED_TYPES_IN_TYPE_DECLARATION
```

```
 FORMATTER / Option for alignment of permitted types in type declaration
     - option id:         "org.eclipse.jdt.core.formatter.alignment_for_permitted_types_in_type_declaration"
     - possible values:   values returned by createAlignmentValue(boolean, int, int) call
     - default:           createAlignmentValue(false, WRAP_COMPACT, INDENT_DEFAULT)
```

Since:

3.33

See Also:

createAlignmentValue(boolean, int, int),
Constant Field Values

FORMATTER_ALIGNMENT_FOR_RECORD_COMPONENTS

```
public static final String  FORMATTER_ALIGNMENT_FOR_RECORD_COMPONENTS
```

```
 FORMATTER / Option for alignment of components in record declaration
     - option id:         "org.eclipse.jdt.core.formatter.alignment_for_record_components"
     - possible values:   values returned by createAlignmentValue(boolean, int, int) call
     - default:           createAlignmentValue(false, WRAP_COMPACT, INDENT_DEFAULT)
```

Since:

3.22

See Also:

createAlignmentValue(boolean, int, int),
Constant Field Values

FORMATTER_ALIGNMENT_FOR_RESOURCES_IN_TRY

```
public static final String  FORMATTER_ALIGNMENT_FOR_RESOURCES_IN_TRY
```

```
 FORMATTER / Option for alignment of resources in a try with resources statement
     - option id:         "org.eclipse.jdt.core.formatter.alignment_for_resources_in_try"
     - possible values:   values returned by createAlignmentValue(boolean, int, int) call
     - default:           createAlignmentValue(false, WRAP_NEXT_PER_LINE, INDENT_DEFAULT)
```

Since:

3.7.1

See Also:

createAlignmentValue(boolean, int, int),
Constant Field Values

FORMATTER_ALIGNMENT_FOR_SELECTOR_IN_METHOD_INVOCATION

```
public static final String  FORMATTER_ALIGNMENT_FOR_SELECTOR_IN_METHOD_INVOCATION
```

```
 FORMATTER / Option for alignment of selector in method invocation
     - option id:         "org.eclipse.jdt.core.formatter.alignment_for_selector_in_method_invocation"
     - possible values:   values returned by createAlignmentValue(boolean, int, int) call
     - default:           createAlignmentValue(false, WRAP_COMPACT, INDENT_DEFAULT)
```

Since:

3.0

See Also:

createAlignmentValue(boolean, int, int),
Constant Field Values

FORMATTER_ALIGNMENT_FOR_SUPERCLASS_IN_TYPE_DECLARATION

```
public static final String  FORMATTER_ALIGNMENT_FOR_SUPERCLASS_IN_TYPE_DECLARATION
```

```
 FORMATTER / Option for alignment of superclass in type declaration
     - option id:         "org.eclipse.jdt.core.formatter.alignment_for_superclass_in_type_declaration"
     - possible values:   values returned by createAlignmentValue(boolean, int, int) call
     - default:           createAlignmentValue(false, WRAP_NEXT_SHIFTED, INDENT_DEFAULT)
```

Since:

3.0

See Also:

createAlignmentValue(boolean, int, int),

Constant Field Values

FORMATTER_ALIGNMENT_FOR_SUPERINTERFACES_IN_ENUM_DECLARATION

```
public static final String  FORMATTER_ALIGNMENT_FOR_SUPERINTERFACES_IN_ENUM_DECLARATION
```

```
 FORMATTER / Option for alignment of superinterfaces in enum declaration
     - option id:         "org.eclipse.jdt.core.formatter.alignment_for_superinterfaces_in_enum_declaration"
     - possible values:   values returned by createAlignmentValue(boolean, int, int) call
     - default:           createAlignmentValue(false, WRAP_COMPACT, INDENT_DEFAULT)
```

Since:

3.1

See Also:

createAlignmentValue(boolean, int, int),
Constant Field Values

FORMATTER_ALIGNMENT_FOR_SUPERINTERFACES_IN_RECORD_DECLARATION

```
public static final String  FORMATTER_ALIGNMENT_FOR_SUPERINTERFACES_IN_RECORD_DECLARATION
```

```
 FORMATTER / Option for alignment of superinterfaces in record declaration
     - option id:         "org.eclipse.jdt.core.formatter.alignment_for_superinterfaces_in_record_declaration"
     - possible values:   values returned by createAlignmentValue(boolean, int, int) call
     - default:           createAlignmentValue(false, WRAP_COMPACT, INDENT_DEFAULT)
```

Since:

3.22

See Also:

createAlignmentValue(boolean, int, int),
Constant Field Values

FORMATTER_ALIGNMENT_FOR_SUPERINTERFACES_IN_TYPE_DECLARATION

```
public static final String  FORMATTER_ALIGNMENT_FOR_SUPERINTERFACES_IN_TYPE_DECLARATION
```

```
 FORMATTER / Option for alignment of superinterfaces in type declaration
     - option id:         "org.eclipse.jdt.core.formatter.alignment_for_superinterfaces_in_type_declaration"
     - possible values:   values returned by createAlignmentValue(boolean, int, int) call
     - default:           createAlignmentValue(false, WRAP_COMPACT, INDENT_DEFAULT)
```

Since:

3.0

See Also:

createAlignmentValue(boolean, int, int),
Constant Field Values

FORMATTER_ALIGNMENT_FOR_SWITCH_CASE_WITH_ARROW

```
public static final String  FORMATTER_ALIGNMENT_FOR_SWITCH_CASE_WITH_ARROW
```

```
 FORMATTER / Option for alignment of arrow in switch case (->)
     - option id:         "org.eclipse.jdt.core.formatter.alignment_for_switch_case_with_arrow"
     - possible values:   values returned by createAlignmentValue(boolean, int, int) call
     - default:           createAlignmentValue(false, WRAP_COMPACT, INDENT_BY_ONE)
```

Since:

3.29

See Also:

createAlignmentValue(boolean, int, int),
Constant Field Values

FORMATTER_ALIGNMENT_FOR_THROWS_CLAUSE_IN_CONSTRUCTOR_DECLARATION

```
public static final String  FORMATTER_ALIGNMENT_FOR_THROWS_CLAUSE_IN_CONSTRUCTOR_DECLARATION
```

```
 FORMATTER / Option for alignment of throws clause in constructor declaration
     - option id:         "org.eclipse.jdt.core.formatter.alignment_for_throws_clause_in_constructor_declaration"
     - possible values:   values returned by createAlignmentValue(boolean, int, int) call
     - default:           createAlignmentValue(false, WRAP_COMPACT, INDENT_DEFAULT)
```

Since:

3.0

See Also:

createAlignmentValue(boolean, int, int),
Constant Field Values

FORMATTER_ALIGNMENT_FOR_THROWS_CLAUSE_IN_METHOD_DECLARATION

```
public static final String  FORMATTER_ALIGNMENT_FOR_THROWS_CLAUSE_IN_METHOD_DECLARATION
```

```
 FORMATTER / Option for alignment of throws clause in method declaration
     - option id:         "org.eclipse.jdt.core.formatter.alignment_for_throws_clause_in_method_declaration"
     - possible values:   values returned by createAlignmentValue(boolean, int, int) call
     - default:           createAlignmentValue(false, WRAP_COMPACT, INDENT_DEFAULT)
```

Since:

3.0

See Also:

createAlignmentValue(boolean, int, int),
Constant Field Values

FORMATTER_ALIGNMENT_FOR_TYPE_ARGUMENTS

```
public static final String  FORMATTER_ALIGNMENT_FOR_TYPE_ARGUMENTS
```

```
 FORMATTER / Option for alignment of type arguments in method invocations and references
     - option id:         "org.eclipse.jdt.core.formatter.alignment_for_type_arguments"
     - possible values:   values returned by createAlignmentValue(boolean, int, int) call
     - default:           createAlignmentValue(false, WRAP_NO_SPLIT, INDENT_DEFAULT)
```

Since:

3.12

See Also:

createAlignmentValue(boolean, int, int),
Constant Field Values

FORMATTER_ALIGNMENT_FOR_TYPE_PARAMETERS

```
public static final String  FORMATTER_ALIGNMENT_FOR_TYPE_PARAMETERS
```

```
 FORMATTER / Option for alignment of type parameters in method and type declarations
     - option id:         "org.eclipse.jdt.core.formatter.alignment_for_type_parameters"
     - possible values:   values returned by createAlignmentValue(boolean, int, int) call
     - default:           createAlignmentValue(false, WRAP_NO_SPLIT, INDENT_DEFAULT)
```

Since:

3.12

See Also:

createAlignmentValue(boolean, int, int),
Constant Field Values

FORMATTER_ALIGNMENT_FOR_UNION_TYPE_IN_MULTICATCH

```
public static final String  FORMATTER_ALIGNMENT_FOR_UNION_TYPE_IN_MULTICATCH
```

```
 FORMATTER / Option for alignment of exceptions declared in a Union Type in the argument of a multicatch statement
     - option id:         "org.eclipse.jdt.core.formatter.alignment_for_union_type_in_multicatch"
     - possible values:   values returned by createAlignmentValue(boolean, int, int) call
     - default:           createAlignmentValue(false, WRAP_COMPACT, INDENT_DEFAULT)
```

Since:

3.7.1

See Also:

createAlignmentValue(boolean, int, int),
Constant Field Values

FORMATTER_BLANK_LINES_AFTER_IMPORTS

```
public static final String  FORMATTER_BLANK_LINES_AFTER_IMPORTS
```

```
 FORMATTER / Option to add or remove blank lines after the imports declaration
     - option id:         "org.eclipse.jdt.core.formatter.blank_lines_after_imports"
     - possible values:   "<n>", where n is an integer. If n is negative, the actual number of
                          blank lines is ~n and any excess blank lines are deleted, overriding the
                          FORMATTER_NUMBER_OF_EMPTY_LINES_TO_PRESERVE option
     - default:           "0"
```

Since:

3.0

See Also:

Constant Field Values

FORMATTER_BLANK_LINES_AFTER_PACKAGE

```
public static final String  FORMATTER_BLANK_LINES_AFTER_PACKAGE
```

```
 FORMATTER / Option to add or remove blank lines after the package declaration
     - option id:         "org.eclipse.jdt.core.formatter.blank_lines_after_package"
     - possible values:   "<n>", where n is an integer. If n is negative, the actual number of
                          blank lines is ~n and any excess blank lines are deleted, overriding the
                          FORMATTER_NUMBER_OF_EMPTY_LINES_TO_PRESERVE option
     - default:           "0"
```

Since:

3.0

See Also:

Constant Field Values

FORMATTER_BLANK_LINES_AT_BEGINNING_OF_METHOD_BODY

```
public static final String  FORMATTER_BLANK_LINES_AT_BEGINNING_OF_METHOD_BODY
```

```
 FORMATTER / Option to add or remove blank lines at the beginning of the method body
     - option id:         "org.eclipse.jdt.core.formatter.number_of_blank_lines_at_beginning_of_method_body"
     - possible values:   "<n>", where n is an integer. If n is negative, the actual number of
                          blank lines is ~n and any excess blank lines are deleted, overriding the
                          FORMATTER_NUMBER_OF_EMPTY_LINES_TO_PRESERVE option
     - default:           "0"
```

Since:

3.0

See Also:

Constant Field Values

FORMATTER_BLANK_LINES_AT_END_OF_METHOD_BODY

```
public static final String  FORMATTER_BLANK_LINES_AT_END_OF_METHOD_BODY
```

```
 FORMATTER / Option to add or remove blank lines at the end of the method body
     - option id:         "org.eclipse.jdt.core.formatter.number_of_blank_lines_at_end_of_method_body"
     - possible values:   "<n>", where n is an integer. If n is negative, the actual number of
                          blank lines is ~n and any excess blank lines are deleted, overriding the
                          FORMATTER_NUMBER_OF_EMPTY_LINES_TO_PRESERVE option
     - default:           "0"
```

Since:

3.19

See Also:

Constant Field Values

FORMATTER_BLANK_LINES_AT_BEGINNING_OF_CODE_BLOCK

```
public static final String  FORMATTER_BLANK_LINES_AT_BEGINNING_OF_CODE_BLOCK
```

```
 FORMATTER / Option to add or remove blank lines at the beginning of the code block
     - option id:         "org.eclipse.jdt.core.formatter.number_of_blank_lines_at_beginning_of_code_block"
     - possible values:   "<n>", where n is an integer. If n is negative, the actual number of
                          blank lines is ~n and any excess blank lines are deleted, overriding the
                          FORMATTER_NUMBER_OF_EMPTY_LINES_TO_PRESERVE option
     - default:           "0"
```

Since:

3.19

See Also:

Constant Field Values

FORMATTER_BLANK_LINES_AT_END_OF_CODE_BLOCK

```
public static final String  FORMATTER_BLANK_LINES_AT_END_OF_CODE_BLOCK
```

```
 FORMATTER / Option to add or remove blank lines at the end of the code block
     - option id:         "org.eclipse.jdt.core.formatter.number_of_blank_lines_at_end_of_code_block"
     - possible values:   "<n>", where n is an integer. If n is negative, the actual number of
                          blank lines is ~n and any excess blank lines are deleted, overriding the
                          FORMATTER_NUMBER_OF_EMPTY_LINES_TO_PRESERVE option
     - default:           "0"
```

Since:

3.19

See Also:

Constant Field Values

FORMATTER_BLANK_LINES_BEFORE_CODE_BLOCK

```
public static final String  FORMATTER_BLANK_LINES_BEFORE_CODE_BLOCK
```

```
 FORMATTER / Option to add or remove blank lines before a statement containing a code block
     - option id:         "org.eclipse.jdt.core.formatter.number_of_blank_lines_before_code_block"
     - possible values:   "<n>", where n is an integer. If n is negative, the actual number of
                          blank lines is ~n and any excess blank lines are deleted, overriding the
                          FORMATTER_NUMBER_OF_EMPTY_LINES_TO_PRESERVE option
     - default:           "0"
```

Since:

3.19

See Also:

Constant Field Values

FORMATTER_BLANK_LINES_AFTER_CODE_BLOCK

```
public static final String  FORMATTER_BLANK_LINES_AFTER_CODE_BLOCK
```

```
 FORMATTER / Option to add or remove blank lines after a statement containing a code block
     - option id:         "org.eclipse.jdt.core.formatter.number_of_blank_lines_after_code_block"
     - possible values:   "<n>", where n is an integer. If n is negative, the actual number of
                          blank lines is ~n and any excess blank lines are deleted, overriding the
                          FORMATTER_NUMBER_OF_EMPTY_LINES_TO_PRESERVE option
     - default:           "0"
```

Since:

3.19

See Also:

Constant Field Values

FORMATTER_BLANK_LINES_BEFORE_FIELD

```
public static final String  FORMATTER_BLANK_LINES_BEFORE_FIELD
```

```
 FORMATTER / Option to add or remove blank lines before a field declaration
     - option id:         "org.eclipse.jdt.core.formatter.blank_lines_before_field"
     - possible values:   "<n>", where n is an integer. If n is negative, the actual number of
                          blank lines is ~n and any excess blank lines are deleted, overriding the
                          FORMATTER_NUMBER_OF_EMPTY_LINES_TO_PRESERVE option
     - default:           "0"
```

Since:

3.0

See Also:

Constant Field Values

FORMATTER_BLANK_LINES_BEFORE_FIRST_CLASS_BODY_DECLARATION

```
public static final String  FORMATTER_BLANK_LINES_BEFORE_FIRST_CLASS_BODY_DECLARATION
```

```
 FORMATTER / Option to add or remove blank lines before the first class body declaration
     - option id:         "org.eclipse.jdt.core.formatter.blank_lines_before_first_class_body_declaration"
     - possible values:   "<n>", where n is an integer. If n is negative, the actual number of
                          blank lines is ~n and any excess blank lines are deleted, overriding the
                          FORMATTER_NUMBER_OF_EMPTY_LINES_TO_PRESERVE option
     - default:           "0"
```

Since:

3.0

See Also:

Constant Field Values

FORMATTER_BLANK_LINES_AFTER_LAST_CLASS_BODY_DECLARATION

```
public static final String  FORMATTER_BLANK_LINES_AFTER_LAST_CLASS_BODY_DECLARATION
```

```
 FORMATTER / Option to add or remove blank lines after the last class body declaration
     - option id:         "org.eclipse.jdt.core.formatter.blank_lines_after_last_class_body_declaration"
     - possible values:   "<n>", where n is an integer. If n is negative, the actual number of
                          blank lines is ~n and any excess blank lines are deleted, overriding the
                          FORMATTER_NUMBER_OF_EMPTY_LINES_TO_PRESERVE option
     - default:           "0"
```

Since:

3.19

See Also:

Constant Field Values

FORMATTER_BLANK_LINES_BEFORE_IMPORTS

```
public static final String  FORMATTER_BLANK_LINES_BEFORE_IMPORTS
```

```
 FORMATTER / Option to add or remove blank lines before the imports declaration
     - option id:         "org.eclipse.jdt.core.formatter.blank_lines_before_imports"
     - possible values:   "<n>", where n is an integer. If n is negative, the actual number of
                          blank lines is ~n and any excess blank lines are deleted, overriding the
                          FORMATTER_NUMBER_OF_EMPTY_LINES_TO_PRESERVE option
     - default:           "0"
```

Since:

3.0

See Also:

Constant Field Values

FORMATTER_BLANK_LINES_BEFORE_ABSTRACT_METHOD

```
public static final String  FORMATTER_BLANK_LINES_BEFORE_ABSTRACT_METHOD
```

```
 FORMATTER / Option to add or remove blank lines before an abstract method declaration
     - option id:         "org.eclipse.jdt.core.formatter.blank_lines_before_abstract_method"
     - possible values:   "<n>", where n is an integer. If n is negative, the actual number of
                          blank lines is ~n and any excess blank lines are deleted, overriding the
                          FORMATTER_NUMBER_OF_EMPTY_LINES_TO_PRESERVE option
     - default:           "0"
```

Since:

3.19

See Also:

Constant Field Values

FORMATTER_BLANK_LINES_BEFORE_MEMBER_TYPE

```
public static final String  FORMATTER_BLANK_LINES_BEFORE_MEMBER_TYPE
```

```
 FORMATTER / Option to add or remove blank lines before a member type declaration
     - option id:         "org.eclipse.jdt.core.formatter.blank_lines_before_member_type"
     - possible values:   "<n>", where n is an integer. If n is negative, the actual number of
                          blank lines is ~n and any excess blank lines are deleted, overriding the
                          FORMATTER_NUMBER_OF_EMPTY_LINES_TO_PRESERVE option
     - default:           "0"
```

Since:

3.0

See Also:

Constant Field Values

FORMATTER_BLANK_LINES_BEFORE_METHOD

```
public static final String  FORMATTER_BLANK_LINES_BEFORE_METHOD
```

```
 FORMATTER / Option to add or remove blank lines before a non-abstract method declaration
     - option id:         "org.eclipse.jdt.core.formatter.blank_lines_before_method"
     - possible values:   "<n>", where n is an integer. If n is negative, the actual number of
                          blank lines is ~n and any excess blank lines are deleted, overriding the
                          FORMATTER_NUMBER_OF_EMPTY_LINES_TO_PRESERVE option
     - default:           "0"
```

Since:

3.0

See Also:

Constant Field Values

FORMATTER_BLANK_LINES_BEFORE_NEW_CHUNK

```
public static final String  FORMATTER_BLANK_LINES_BEFORE_NEW_CHUNK
```

```
 FORMATTER / Option to add or remove blank lines before a new chunk
     - option id:         "org.eclipse.jdt.core.formatter.blank_lines_before_new_chunk"
     - possible values:   "<n>", where n is an integer. If n is negative, the actual number of
                          blank lines is ~n and any excess blank lines are deleted, overriding the
                          FORMATTER_NUMBER_OF_EMPTY_LINES_TO_PRESERVE option
     - default:           "0"
```

Since:

3.0

See Also:

Constant Field Values

FORMATTER_BLANK_LINES_BEFORE_PACKAGE

```
public static final String  FORMATTER_BLANK_LINES_BEFORE_PACKAGE
```

```
 FORMATTER / Option to add or remove blank lines before the package declaration
     - option id:         "org.eclipse.jdt.core.formatter.blank_lines_before_package"
     - possible values:   "<n>", where n is an integer. If n is negative, the actual number of
                          blank lines is ~n and any excess blank lines are deleted, overriding the
                          FORMATTER_NUMBER_OF_EMPTY_LINES_TO_PRESERVE option
     - default:           "0"
```

Since:

3.0

See Also:

Constant Field Values

FORMATTER_BLANK_LINES_BETWEEN_IMPORT_GROUPS

```
public static final String  FORMATTER_BLANK_LINES_BETWEEN_IMPORT_GROUPS
```

```
 FORMATTER / Option to add or remove blank lines between import groups
     - option id:         "org.eclipse.jdt.core.formatter.blank_lines_between_import_groups"
     - possible values:   "<n>", where n is an integer. If n is negative, the actual number of
                          blank lines is ~n and any excess blank lines are deleted, overriding the
                          FORMATTER_NUMBER_OF_EMPTY_LINES_TO_PRESERVE option
     - default:           "1"
```

Note: Import groups are defined once "Organize Import" operation has been executed. The code formatter itself doesn't define the import groups.

Since:

3.3

See Also:

Constant Field Values

FORMATTER_BLANK_LINES_BETWEEN_TYPE_DECLARATIONS

```
public static final String  FORMATTER_BLANK_LINES_BETWEEN_TYPE_DECLARATIONS
```

```
 FORMATTER / Option to add or remove blank lines between type declarations
     - option id:         "org.eclipse.jdt.core.formatter.blank_lines_between_type_declarations"
     - possible values:   "<n>", where n is an integer. If n is negative, the actual number of
                          blank lines is ~n and any excess blank lines are deleted, overriding the
                          FORMATTER_NUMBER_OF_EMPTY_LINES_TO_PRESERVE option
     - default:           "0"
```

Since:

3.0

See Also:

Constant Field Values

FORMATTER_BLANK_LINES_BETWEEN_STATEMENT_GROUPS_IN_SWITCH

```
public static final String  FORMATTER_BLANK_LINES_BETWEEN_STATEMENT_GROUPS_IN_SWITCH
```

```
 FORMATTER / Option to add or remove blank lines between statement groups in switch
     - option id:         "org.eclipse.jdt.core.formatter.blank_lines_between_statement_groups_in_switch"
     - possible values:   "<n>", where n is an integer. If n is negative, the actual number of
                          blank lines is ~n and any excess blank lines are deleted, overriding the
                          FORMATTER_NUMBER_OF_EMPTY_LINES_TO_PRESERVE option
     - default:           "0"
```

Since:

3.19

See Also:

Constant Field Values

FORMATTER_BRACE_POSITION_FOR_ANNOTATION_TYPE_DECLARATION

```
public static final String  FORMATTER_BRACE_POSITION_FOR_ANNOTATION_TYPE_DECLARATION
```

```
 FORMATTER / Option to position the braces of an annotation type declaration
     - option id:         "org.eclipse.jdt.core.formatter.brace_position_for_annotation_type_declaration"
     - possible values:   { END_OF_LINE, NEXT_LINE, NEXT_LINE_SHIFTED, NEXT_LINE_ON_WRAP }
     - default:           END_OF_LINE
```

Since:

3.1

See Also:

END_OF_LINE, NEXT_LINE, NEXT_LINE_SHIFTED, NEXT_LINE_ON_WRAP, Constant Field Values

FORMATTER_BRACE_POSITION_FOR_ANONYMOUS_TYPE_DECLARATION

```
public static final String  FORMATTER_BRACE_POSITION_FOR_ANONYMOUS_TYPE_DECLARATION
```

```
 FORMATTER / Option to position the braces of an anonymous type declaration
     - option id:         "org.eclipse.jdt.core.formatter.brace_position_for_anonymous_type_declaration"
     - possible values:   { END_OF_LINE, NEXT_LINE, NEXT_LINE_SHIFTED, NEXT_LINE_ON_WRAP }
     - default:           END_OF_LINE
```

Since:

3.0

See Also:

END_OF_LINE, NEXT_LINE, NEXT_LINE_SHIFTED, NEXT_LINE_ON_WRAP, Constant Field Values

FORMATTER_BRACE_POSITION_FOR_ARRAY_INITIALIZER

```
public static final String  FORMATTER_BRACE_POSITION_FOR_ARRAY_INITIALIZER
```

```
 FORMATTER / Option to position the braces of an array initializer
     - option id:         "org.eclipse.jdt.core.formatter.brace_position_for_array_initializer"
     - possible values:   { END_OF_LINE, NEXT_LINE, NEXT_LINE_SHIFTED, NEXT_LINE_ON_WRAP }
     - default:           END_OF_LINE
```

Since:

3.0

See Also:

END_OF_LINE, NEXT_LINE, NEXT_LINE_SHIFTED, NEXT_LINE_ON_WRAP, Constant Field Values

FORMATTER_BRACE_POSITION_FOR_BLOCK

```
public static final String  FORMATTER_BRACE_POSITION_FOR_BLOCK
```

```
 FORMATTER / Option to position the braces of a block
     - option id:         "org.eclipse.jdt.core.formatter.brace_position_for_block"
     - possible values:   { END_OF_LINE, NEXT_LINE, NEXT_LINE_SHIFTED, NEXT_LINE_ON_WRAP }
     - default:           END_OF_LINE
```

Since:

3.0

See Also:

END_OF_LINE, NEXT_LINE, NEXT_LINE_SHIFTED, NEXT_LINE_ON_WRAP, Constant Field Values

FORMATTER_BRACE_POSITION_FOR_BLOCK_IN_CASE

```
public static final String  FORMATTER_BRACE_POSITION_FOR_BLOCK_IN_CASE
```

```
 FORMATTER / Option to position the braces of a block in a switch statement/expression when the block
             is the first statement following a case with colon
     - option id:         "org.eclipse.jdt.core.formatter.brace_position_for_block_in_case"
     - possible values:   { END_OF_LINE, NEXT_LINE, NEXT_LINE_SHIFTED, NEXT_LINE_ON_WRAP }
     - default:           END_OF_LINE
```

Since:

3.0

See Also:

END_OF_LINE, NEXT_LINE, NEXT_LINE_SHIFTED, NEXT_LINE_ON_WRAP, Constant Field Values

FORMATTER_BRACE_POSITION_FOR_BLOCK_IN_CASE_AFTER_ARROW

```
public static final String  FORMATTER_BRACE_POSITION_FOR_BLOCK_IN_CASE_AFTER_ARROW
```

```
 FORMATTER / Option to position the braces of a block in a switch statement/expression when the block
             is the first statement following a case with arrow
     - option id:         "org.eclipse.jdt.core.formatter.brace_position_for_block_in_case_after_arrow"
     - possible values:   { END_OF_LINE, NEXT_LINE, NEXT_LINE_SHIFTED, NEXT_LINE_ON_WRAP }
     - default:           END_OF_LINE
```

Since:

3.37

See Also:

END_OF_LINE, NEXT_LINE, NEXT_LINE_SHIFTED, NEXT_LINE_ON_WRAP, Constant Field Values

FORMATTER_BRACE_POSITION_FOR_CONSTRUCTOR_DECLARATION

```
public static final String  FORMATTER_BRACE_POSITION_FOR_CONSTRUCTOR_DECLARATION
```

```
 FORMATTER / Option to position the braces of a constructor declaration
     - option id:         "org.eclipse.jdt.core.formatter.brace_position_for_constructor_declaration"
```

```
     - possible values:   { END_OF_LINE, NEXT_LINE, NEXT_LINE_SHIFTED, NEXT_LINE_ON_WRAP }
     - default:           END_OF_LINE
```

Since:

3.0

See Also:

END_OF_LINE, NEXT_LINE, NEXT_LINE_SHIFTED, NEXT_LINE_ON_WRAP, Constant Field Values

FORMATTER_BRACE_POSITION_FOR_ENUM_CONSTANT

```
public static final String  FORMATTER_BRACE_POSITION_FOR_ENUM_CONSTANT
```

```
 FORMATTER / Option to position the braces of an enum constant
     - option id:         "org.eclipse.jdt.core.formatter.brace_position_for_enum_constant"
     - possible values:   { END_OF_LINE, NEXT_LINE, NEXT_LINE_SHIFTED, NEXT_LINE_ON_WRAP }
     - default:           END_OF_LINE
```

Since:

3.1

See Also:

END_OF_LINE, NEXT_LINE, NEXT_LINE_SHIFTED, NEXT_LINE_ON_WRAP, Constant Field Values

FORMATTER_BRACE_POSITION_FOR_ENUM_DECLARATION

```
public static final String  FORMATTER_BRACE_POSITION_FOR_ENUM_DECLARATION
```

```
 FORMATTER / Option to position the braces of an enum declaration
     - option id:         "org.eclipse.jdt.core.formatter.brace_position_for_enum_declaration"
     - possible values:   { END_OF_LINE, NEXT_LINE, NEXT_LINE_SHIFTED, NEXT_LINE_ON_WRAP }
     - default:           END_OF_LINE
```

Since:

3.1

See Also:

END_OF_LINE, NEXT_LINE, NEXT_LINE_SHIFTED, NEXT_LINE_ON_WRAP, Constant Field Values

FORMATTER_BRACE_POSITION_FOR_METHOD_DECLARATION

```
public static final String  FORMATTER_BRACE_POSITION_FOR_METHOD_DECLARATION
```

```
 FORMATTER / Option to position the braces of a method declaration
     - option id:         "org.eclipse.jdt.core.formatter.brace_position_for_method_declaration"
     - possible values:   { END_OF_LINE, NEXT_LINE, NEXT_LINE_SHIFTED, NEXT_LINE_ON_WRAP }
     - default:           END_OF_LINE
```

Since:

3.0

See Also:

END_OF_LINE, NEXT_LINE, NEXT_LINE_SHIFTED, NEXT_LINE_ON_WRAP, Constant Field Values

FORMATTER_BRACE_POSITION_FOR_RECORD_CONSTRUCTOR

```
public static final String  FORMATTER_BRACE_POSITION_FOR_RECORD_CONSTRUCTOR
```

```
 FORMATTER / Option to position the braces of a record constructor
     - option id:         "org.eclipse.jdt.core.formatter.brace_position_for_record_constructor"
     - possible values:   { END_OF_LINE, NEXT_LINE, NEXT_LINE_SHIFTED, NEXT_LINE_ON_WRAP }
     - default:           END_OF_LINE
```

Since:

3.22

See Also:

END_OF_LINE, NEXT_LINE, NEXT_LINE_SHIFTED, NEXT_LINE_ON_WRAP, Constant Field Values

FORMATTER_BRACE_POSITION_FOR_RECORD_DECLARATION

```
public static final String  FORMATTER_BRACE_POSITION_FOR_RECORD_DECLARATION
```

```
 FORMATTER / Option to position the braces of a record declaration
     - option id:         "org.eclipse.jdt.core.formatter.brace_position_for_record_declaration"
```

```
     - possible values:   { END_OF_LINE, NEXT_LINE, NEXT_LINE_SHIFTED, NEXT_LINE_ON_WRAP }
     - default:           END_OF_LINE
```

Since:

3.22

See Also:

END_OF_LINE, NEXT_LINE, NEXT_LINE_SHIFTED, NEXT_LINE_ON_WRAP, Constant Field Values

FORMATTER_BRACE_POSITION_FOR_SWITCH

```
public static final String  FORMATTER_BRACE_POSITION_FOR_SWITCH
```

```
 FORMATTER / Option to position the braces of a switch statement
     - option id:         "org.eclipse.jdt.core.formatter.brace_position_for_switch"
     - possible values:   { END_OF_LINE, NEXT_LINE, NEXT_LINE_SHIFTED, NEXT_LINE_ON_WRAP }
     - default:           END_OF_LINE
```

Since:

3.0

See Also:

END_OF_LINE, NEXT_LINE, NEXT_LINE_SHIFTED, NEXT_LINE_ON_WRAP, Constant Field Values

FORMATTER_BRACE_POSITION_FOR_TYPE_DECLARATION

```
public static final String  FORMATTER_BRACE_POSITION_FOR_TYPE_DECLARATION
```

```
 FORMATTER / Option to position the braces of a type declaration
     - option id:         "org.eclipse.jdt.core.formatter.brace_position_for_type_declaration"
     - possible values:   { END_OF_LINE, NEXT_LINE, NEXT_LINE_SHIFTED, NEXT_LINE_ON_WRAP }
     - default:           END_OF_LINE
```

Since:

3.0

See Also:

END_OF_LINE, NEXT_LINE, NEXT_LINE_SHIFTED, NEXT_LINE_ON_WRAP, Constant Field Values

FORMATTER_BRACE_POSITION_FOR_LAMBDA_BODY

```
public static final String  FORMATTER_BRACE_POSITION_FOR_LAMBDA_BODY
```

```
 FORMATTER / Option to position the braces of a lambda block
     - option id:         "org.eclipse.jdt.core.formatter.brace_position_for_lambda_body"
     - possible values:   { END_OF_LINE, NEXT_LINE, NEXT_LINE_SHIFTED, NEXT_LINE_ON_WRAP }
     - default:           END_OF_LINE
```

Since:

3.10

See Also:

END_OF_LINE, NEXT_LINE, NEXT_LINE_SHIFTED, NEXT_LINE_ON_WRAP, Constant Field Values

FORMATTER_PARENTHESES_POSITIONS_IN_METHOD_DECLARATION

```
public static final String  FORMATTER_PARENTHESES_POSITIONS_IN_METHOD_DECLARATION
```

```
 FORMATTER / Option to position parentheses in method declarations
     - option id:         "org.eclipse.jdt.core.formatter.parentheses_positions_in_method_delcaration"
     - possible values:   { COMMON_LINES, SEPARATE_LINES_IF_NOT_EMPTY, SEPARATE_LINES_IF_WRAPPED, SEPARATE_LINES, PRESERVE_POSITIONS }
     - default:           COMMON_LINES
```

Note that there is a typo (`delcaration` vs. `declaration`) in the option name. It has to remain, as fixing it would be a breaking change.

Since:

3.12

See Also:

COMMON_LINES, SEPARATE_LINES_IF_NOT_EMPTY, SEPARATE_LINES_IF_WRAPPED, SEPARATE_LINES, PRESERVE_POSITIONS, Constant Field Values

FORMATTER_PARENTHESES_POSITIONS_IN_METHOD_INVOCATION

```
public static final String  FORMATTER_PARENTHESES_POSITIONS_IN_METHOD_INVOCATION
```

```
 FORMATTER / Option to position parentheses in method invocations
     - option id:         "org.eclipse.jdt.core.formatter.parentheses_positions_in_method_invocation"
     - possible values:   { COMMON_LINES, SEPARATE_LINES_IF_NOT_EMPTY, SEPARATE_LINES_IF_WRAPPED, SEPARATE_LINES, PRESERVE_POSITIONS }
     - default:           COMMON_LINES
```

Since:

3.12

See Also:

COMMON_LINES, SEPARATE_LINES_IF_NOT_EMPTY, SEPARATE_LINES_IF_WRAPPED, SEPARATE_LINES, PRESERVE_POSITIONS, Constant Field Values

FORMATTER_PARENTHESES_POSITIONS_IN_ENUM_CONSTANT_DECLARATION

```
public static final String  FORMATTER_PARENTHESES_POSITIONS_IN_ENUM_CONSTANT_DECLARATION
```

```
 FORMATTER / Option to position parentheses in enum constant declarations
     - option id:         "org.eclipse.jdt.core.formatter.parentheses_positions_in_enum_constant_declaration"
     - possible values:   { COMMON_LINES, SEPARATE_LINES_IF_NOT_EMPTY, SEPARATE_LINES_IF_WRAPPED, SEPARATE_LINES, PRESERVE_POSITIONS }
     - default:           COMMON_LINES
```

Since:

3.12

See Also:

COMMON_LINES, SEPARATE_LINES_IF_NOT_EMPTY, SEPARATE_LINES_IF_WRAPPED, SEPARATE_LINES, PRESERVE_POSITIONS, Constant Field Values

FORMATTER_PARENTHESES_POSITIONS_IN_RECORD_DECLARATION

```
public static final String  FORMATTER_PARENTHESES_POSITIONS_IN_RECORD_DECLARATION
```

```
 FORMATTER / Option to position parentheses in record declarations
     - option id:         "org.eclipse.jdt.core.formatter.parentheses_positions_in_record_declaration"
     - possible values:   { COMMON_LINES, SEPARATE_LINES_IF_NOT_EMPTY, SEPARATE_LINES_IF_WRAPPED, SEPARATE_LINES, PRESERVE_POSITIONS }
     - default:           COMMON_LINES
```

Since:

3.22

See Also:

COMMON_LINES, SEPARATE_LINES_IF_NOT_EMPTY, SEPARATE_LINES_IF_WRAPPED, SEPARATE_LINES, PRESERVE_POSITIONS, Constant Field Values

FORMATTER_PARENTHESES_POSITIONS_IN_IF_WHILE_STATEMENT

```
public static final String  FORMATTER_PARENTHESES_POSITIONS_IN_IF_WHILE_STATEMENT
```

```
 FORMATTER / Option to position parentheses in 'if' and 'while' statements
     - option id:         "org.eclipse.jdt.core.formatter.parentheses_positions_in_if_while_statement"
     - possible values:   { COMMON_LINES, SEPARATE_LINES_IF_WRAPPED, SEPARATE_LINES, PRESERVE_POSITIONS }
     - default:           COMMON_LINES
```

Since:

3.12

See Also:

COMMON_LINES, SEPARATE_LINES_IF_WRAPPED, SEPARATE_LINES, PRESERVE_POSITIONS, Constant Field Values

FORMATTER_PARENTHESES_POSITIONS_IN_FOR_STATEMENT

```
public static final String  FORMATTER_PARENTHESES_POSITIONS_IN_FOR_STATEMENT
```

```
 FORMATTER / Option to position parentheses in 'for' statements
     - option id:         "org.eclipse.jdt.core.formatter.parentheses_positions_in_for_statment"
     - possible values:   { COMMON_LINES,  SEPARATE_LINES_IF_WRAPPED, SEPARATE_LINES, PRESERVE_POSITIONS }
     - default:           COMMON_LINES
```

Note that there is a typo (`statment` vs. `statement`) in the option name. It has to remain, as fixing it would be a breaking change.

Since:

3.12

See Also:

COMMON_LINES, SEPARATE_LINES_IF_WRAPPED, SEPARATE_LINES, PRESERVE_POSITIONS, Constant Field Values

FORMATTER_PARENTHESES_POSITIONS_IN_SWITCH_STATEMENT

```
public static final String  FORMATTER_PARENTHESES_POSITIONS_IN_SWITCH_STATEMENT
```

```
 FORMATTER / Option to position parentheses in 'switch' statements
     - option id:         "org.eclipse.jdt.core.formatter.parentheses_positions_in_switch_statement"
     - possible values:   { COMMON_LINES, SEPARATE_LINES_IF_WRAPPED, SEPARATE_LINES, PRESERVE_POSITIONS }
     - default:           COMMON_LINES
```

Since:

3.12

See Also:

COMMON_LINES, SEPARATE_LINES_IF_WRAPPED, SEPARATE_LINES, PRESERVE_POSITIONS, Constant Field Values

FORMATTER_PARENTHESES_POSITIONS_IN_TRY_CLAUSE

```
public static final String  FORMATTER_PARENTHESES_POSITIONS_IN_TRY_CLAUSE
```

```
 FORMATTER / Option to position parentheses in try clauses
     - option id:         "org.eclipse.jdt.core.formatter.parentheses_positions_in_try_clause"
     - possible values:   { COMMON_LINES, SEPARATE_LINES_IF_WRAPPED, SEPARATE_LINES, PRESERVE_POSITIONS }
     - default:           COMMON_LINES
```

Since:

3.12

See Also:

COMMON_LINES, SEPARATE_LINES_IF_WRAPPED, SEPARATE_LINES, PRESERVE_POSITIONS, Constant Field Values

FORMATTER_PARENTHESES_POSITIONS_IN_CATCH_CLAUSE

```
public static final String  FORMATTER_PARENTHESES_POSITIONS_IN_CATCH_CLAUSE
```

```
 FORMATTER / Option to position parentheses in catch clauses
     - option id:         "org.eclipse.jdt.core.formatter.parentheses_positions_in_catch_clause"
     - possible values:   { COMMON_LINES, SEPARATE_LINES_IF_WRAPPED, SEPARATE_LINES, PRESERVE_POSITIONS }
     - default:           COMMON_LINES
```

Since:

3.12

See Also:

COMMON_LINES, SEPARATE_LINES_IF_WRAPPED, SEPARATE_LINES, PRESERVE_POSITIONS, Constant Field Values

FORMATTER_PARENTHESES_POSITIONS_IN_ANNOTATION

```
public static final String  FORMATTER_PARENTHESES_POSITIONS_IN_ANNOTATION
```

```
 FORMATTER / Option to position parentheses in annotations
     - option id:         "org.eclipse.jdt.core.formatter.parentheses_positions_in_annotation"
     - possible values:   { COMMON_LINES, SEPARATE_LINES_IF_NOT_EMPTY, SEPARATE_LINES_IF_WRAPPED, SEPARATE_LINES, PRESERVE_POSITIONS }
     - default:           COMMON_LINES
```

Since:

3.12

See Also:

COMMON_LINES, SEPARATE_LINES_IF_NOT_EMPTY, SEPARATE_LINES_IF_WRAPPED, SEPARATE_LINES, PRESERVE_POSITIONS, Constant Field Values

FORMATTER_PARENTHESES_POSITIONS_IN_LAMBDA_DECLARATION

```
public static final String  FORMATTER_PARENTHESES_POSITIONS_IN_LAMBDA_DECLARATION
```

```
 FORMATTER / Option to position parentheses in lambda declarations
     - option id:         "org.eclipse.jdt.core.formatter.parentheses_positions_in_lambda_declaration"
     - possible values:   { COMMON_LINES, SEPARATE_LINES_IF_NOT_EMPTY, SEPARATE_LINES_IF_WRAPPED, SEPARATE_LINES, PRESERVE_POSITIONS }
     - default:           COMMON_LINES
```

Since:

3.12

See Also:

COMMON_LINES, SEPARATE_LINES_IF_NOT_EMPTY, SEPARATE_LINES_IF_WRAPPED, SEPARATE_LINES, PRESERVE_POSITIONS, Constant Field Values

FORMATTER_COMMENT_CLEAR_BLANK_LINES

```
@Deprecated
public static final String  FORMATTER_COMMENT_CLEAR_BLANK_LINES
```

Deprecated.
Use `FORMATTER_COMMENT_CLEAR_BLANK_LINES_IN_BLOCK_COMMENT` and `FORMATTER_COMMENT_CLEAR_BLANK_LINES_IN_JAVADOC_COMMENT`

```
 FORMATTER / Option to control whether blank lines are cleared inside comments
     - option id:         "org.eclipse.jdt.core.formatter.comment.clear_blank_lines"
     - possible values:   { TRUE, FALSE }
     - default:           FALSE
```

Since:

3.1

See Also:

`TRUE`, `FALSE`, Constant Field Values

FORMATTER_COMMENT_CLEAR_BLANK_LINES_IN_JAVADOC_COMMENT

```
public static final String  FORMATTER_COMMENT_CLEAR_BLANK_LINES_IN_JAVADOC_COMMENT
```

```
 FORMATTER / Option to control whether blank lines are cleared inside javadoc comments
     - option id:         "org.eclipse.jdt.core.formatter.comment.clear_blank_lines_in_javadoc_comment"
     - possible values:   { TRUE, FALSE }
     - default:           FALSE
```

Since:

3.3

See Also:

`TRUE`, `FALSE`, Constant Field Values

FORMATTER_COMMENT_CLEAR_BLANK_LINES_IN_BLOCK_COMMENT

```
public static final String  FORMATTER_COMMENT_CLEAR_BLANK_LINES_IN_BLOCK_COMMENT
```

```
 FORMATTER / Option to control whether blank lines are cleared inside block comments
     - option id:         "org.eclipse.jdt.core.formatter.comment.clear_blank_lines_in_block_comment"
     - possible values:   { TRUE, FALSE }
     - default:           FALSE
```

Since:

3.3

See Also:

`TRUE`, `FALSE`, Constant Field Values

FORMATTER_COMMENT_FORMAT

```
@Deprecated
public static final String  FORMATTER_COMMENT_FORMAT
```

Deprecated.
Use multiple settings for each kind of comments. See `FORMATTER_COMMENT_FORMAT_BLOCK_COMMENT`, `FORMATTER_COMMENT_FORMAT_JAVADOC_COMMENT` and `FORMATTER_COMMENT_FORMAT_LINE_COMMENT`.

```
 FORMATTER / Option to control whether comments are formatted
     - option id:         "org.eclipse.jdt.core.formatter.comment.format_comments"
     - possible values:   { TRUE, FALSE }
     - default:           TRUE
```

Since:

3.1

See Also:

`TRUE`, `FALSE`, Constant Field Values

FORMATTER_COMMENT_FORMAT_LINE_COMMENT

```
public static final String  FORMATTER_COMMENT_FORMAT_LINE_COMMENT
```

```
 FORMATTER / Option to control whether single line comments are formatted
     - option id:         "org.eclipse.jdt.core.formatter.comment.format_line_comments"
     - possible values:   { TRUE, FALSE }
```

```
     - default:           TRUE
```

Since:

3.3

See Also:

`TRUE`, `FALSE`, Constant Field Values

FORMATTER_COMMENT_FORMAT_LINE_COMMENT_STARTING_ON_FIRST_COLUMN

```
public static final String  FORMATTER_COMMENT_FORMAT_LINE_COMMENT_STARTING_ON_FIRST_COLUMN
```

```
 FORMATTER / Option to format line comments that start on the first column
     - option id:         "org.eclipse.jdt.core.formatter.format_line_comment_starting_on_first_column"
     - possible values:   { TRUE, FALSE }
     - default:           TRUE
```

Note that this option is ignored if either the `FORMATTER_COMMENT_FORMAT_LINE_COMMENT` option has been set to `FALSE` or the formatter is created with the mode `ToolFactory.M_FORMAT_NEW`.

Since:

3.6

See Also:

`TRUE`,
`FALSE`,
`ToolFactory.createCodeFormatter(Map, int)`,
Constant Field Values

FORMATTER_COMMENT_PRESERVE_WHITE_SPACE_BETWEEN_CODE_AND_LINE_COMMENT

```
public static final String  FORMATTER_COMMENT_PRESERVE_WHITE_SPACE_BETWEEN_CODE_AND_LINE_COMMENT
```

```
 FORMATTER / Option to control whether the white space between code and line comments should be preserved or replaced with a single space
     - option id:         "org.eclipse.jdt.core.formatter.comment.preserve_white_space_between_code_and_line_comments"
     - possible values:   { TRUE, FALSE }
     - default:           FALSE
```

Since:

3.7

See Also:

`TRUE`, `FALSE`, Constant Field Values

FORMATTER_COMMENT_FORMAT_BLOCK_COMMENT

```
public static final String  FORMATTER_COMMENT_FORMAT_BLOCK_COMMENT
```

```
 FORMATTER / Option to control whether multiple lines comments are formatted
     - option id:         "org.eclipse.jdt.core.formatter.comment.format_block_comments"
     - possible values:   { TRUE, FALSE }
     - default:           TRUE
```

Since:

3.3

See Also:

`TRUE`, `FALSE`, Constant Field Values

FORMATTER_COMMENT_FORMAT_JAVADOC_COMMENT

```
public static final String  FORMATTER_COMMENT_FORMAT_JAVADOC_COMMENT
```

```
 FORMATTER / Option to control whether javadoc comments are formatted
     - option id:         "org.eclipse.jdt.core.formatter.comment.format_javadoc_comments"
     - possible values:   { TRUE, FALSE }
     - default:           TRUE
```

Since:

3.3

See Also:

`TRUE`, `FALSE`, Constant Field Values

FORMATTER_COMMENT_FORMAT_MARKDOWN_COMMENT

```
public static final String  FORMATTER_COMMENT_FORMAT_MARKDOWN_COMMENT
```

```
 FORMATTER / Option to control whether markdown comments are formatted
     - option id:         "org.eclipse.jdt.core.formatter.comment.format_markdown_comments"
     - possible values:   { TRUE, FALSE }
     - default:           TRUE
```

Since:

3.44

See Also:

TRUE, FALSE, Constant Field Values

FORMATTER_COMMENT_FORMAT_HEADER

```
public static final String  FORMATTER_COMMENT_FORMAT_HEADER
```

```
 FORMATTER / Option to control whether the header comment of a Java source file is formatted
     - option id:         "org.eclipse.jdt.core.formatter.comment.format_header"
     - possible values:   { TRUE, FALSE }
     - default:           FALSE
```

Since:

3.1

See Also:

TRUE, FALSE, Constant Field Values

FORMATTER_COMMENT_FORMAT_HTML

```
public static final String  FORMATTER_COMMENT_FORMAT_HTML
```

```
 FORMATTER / Option to control whether HTML tags are formatted.
     - option id:         "org.eclipse.jdt.core.formatter.comment.format_html"
     - possible values:   { TRUE, FALSE }
     - default:           TRUE
```

Since:

3.1

See Also:

TRUE, FALSE, Constant Field Values

FORMATTER_COMMENT_FORMAT_SOURCE

```
public static final String  FORMATTER_COMMENT_FORMAT_SOURCE
```

```
 FORMATTER / Option to control whether code snippets are formatted in comments
     - option id:         "org.eclipse.jdt.core.formatter.comment.format_source_code"
     - possible values:   { TRUE, FALSE }
     - default:           TRUE
```

Since:

3.1

See Also:

TRUE, FALSE, Constant Field Values

FORMATTER_COMMENT_INDENT_PARAMETER_DESCRIPTION

```
public static final String  FORMATTER_COMMENT_INDENT_PARAMETER_DESCRIPTION
```

```
 FORMATTER / Option to control whether description of Javadoc parameters are indented
     - option id:         "org.eclipse.jdt.core.formatter.comment.indent_parameter_description"
     - possible values:   { TRUE, FALSE }
     - default:           TRUE
```

Since:

3.1

See Also:

TRUE,
FALSE,
FORMATTER_COMMENT_INDENT_TAG_DESCRIPTION,
Constant Field Values

FORMATTER_COMMENT_INDENT_TAG_DESCRIPTION

```
public static final String  FORMATTER_COMMENT_INDENT_TAG_DESCRIPTION
```

```
 FORMATTER / Option to control whether Javadoc tag descriptions are indented when wrapped,
     excluding tags controlled by #FORMATTER_COMMENT_INDENT_PARAMETER_DESCRIPTION
     - option id:         "org.eclipse.jdt.core.formatter.comment.indent_return_description"
     - possible values:   { TRUE, FALSE }
     - default:           TRUE
```

Since:

3.17

See Also:

TRUE,
FALSE,
FORMATTER_COMMENT_INDENT_PARAMETER_DESCRIPTION,
Constant Field Values

FORMATTER_COMMENT_INDENT_ROOT_TAGS

```
public static final String  FORMATTER_COMMENT_INDENT_ROOT_TAGS
```

```
 FORMATTER / Option to control whether Javadoc root tags are indented.
     - option id:         "org.eclipse.jdt.core.formatter.comment.indent_root_tags"
     - possible values:   { TRUE, FALSE }
     - default:           TRUE
```

Note that at most one of these options can be set to TRUE:

- FORMATTER_COMMENT_INDENT_ROOT_TAGS,
- FORMATTER_COMMENT_ALIGN_TAGS_NAMES_DESCRIPTIONS,
- FORMATTER_COMMENT_ALIGN_TAGS_DESCREIPTIONS_GROUPED.

Since:

3.1

See Also:

TRUE,
FALSE,
FORMATTER_COMMENT_ALIGN_TAGS_NAMES_DESCRIPTIONS,
FORMATTER_COMMENT_ALIGN_TAGS_DESCREIPTIONS_GROUPED,
Constant Field Values

FORMATTER_COMMENT_ALIGN_TAGS_NAMES_DESCRIPTIONS

```
public static final String  FORMATTER_COMMENT_ALIGN_TAGS_NAMES_DESCRIPTIONS
```

```
 FORMATTER / Option to control whether names and descriptions in Javadoc root tags should be aligned.
     - option id:         "org.eclipse.jdt.core.formatter.comment.align_tags_names_descriptions"
     - possible values:   { TRUE, FALSE }
     - default:           TRUE
```

Note that at most one of these options can be set to TRUE:

- FORMATTER_COMMENT_INDENT_ROOT_TAGS,
- FORMATTER_COMMENT_ALIGN_TAGS_NAMES_DESCRIPTIONS,
- FORMATTER_COMMENT_ALIGN_TAGS_DESCREIPTIONS_GROUPED.

Since:

3.14

See Also:

TRUE,
FALSE,
FORMATTER_COMMENT_INDENT_ROOT_TAGS,
FORMATTER_COMMENT_ALIGN_TAGS_DESCREIPTIONS_GROUPED,
Constant Field Values

FORMATTER_COMMENT_ALIGN_TAGS_DESCREIPTIONS_GROUPED

```
public static final String  FORMATTER_COMMENT_ALIGN_TAGS_DESCREIPTIONS_GROUPED
```

```
 FORMATTER / Option to control whether descriptions and names in Javadoc root tags, should be aligned and grouped by tag type.
     - option id:         "org.eclipse.jdt.core.formatter.comment.align_tags_descriptions_grouped"
     - possible values:   { TRUE, FALSE }
     - default:           TRUE
```

Note that at most one of these options can be set to TRUE:

- FORMATTER_COMMENT_INDENT_ROOT_TAGS,
- FORMATTER_COMMENT_ALIGN_TAGS_NAMES_DESCRIPTIONS,
- FORMATTER_COMMENT_ALIGN_TAGS_DESCREIPTIONS_GROUPED.

Since:

3.14

See Also:

TRUE,
FALSE,
FORMATTER_COMMENT_INDENT_ROOT_TAGS,
FORMATTER_COMMENT_ALIGN_TAGS_NAMES_DESCRIPTIONS,
Constant Field Values

FORMATTER_COMMENT_INSERT_EMPTY_LINE_BEFORE_ROOT_TAGS

```
public static final String  FORMATTER_COMMENT_INSERT_EMPTY_LINE_BEFORE_ROOT_TAGS
```

```
 FORMATTER / Option to insert an empty line before the Javadoc root tag block
     - option id:         "org.eclipse.jdt.core.formatter.comment.insert_new_line_before_root_tags"
     - possible values:   { INSERT, DO_NOT_INSERT }
     - default:           INSERT
```

Since:

3.1

See Also:

JavaCore.INSERT, JavaCore.DO_NOT_INSERT, Constant Field Values

FORMATTER_COMMENT_INSERT_EMPTY_LINE_BETWEEN_DIFFERENT_TAGS

```
public static final String  FORMATTER_COMMENT_INSERT_EMPTY_LINE_BETWEEN_DIFFERENT_TAGS
```

```
 FORMATTER / Option to insert an empty line between Javadoc tags of different type
     - option id:         "org.eclipse.jdt.core.formatter.comment.insert_new_line_between_different_tags"
     - possible values:   { INSERT, DO_NOT_INSERT }
     - default:           INSERT
```

Since:

3.20

See Also:

JavaCore.INSERT, JavaCore.DO_NOT_INSERT, Constant Field Values

FORMATTER_COMMENT_INSERT_NEW_LINE_FOR_PARAMETER

```
public static final String  FORMATTER_COMMENT_INSERT_NEW_LINE_FOR_PARAMETER
```

```
 FORMATTER / Option to insert a new line after Javadoc root tag parameters
     - option id:         "org.eclipse.jdt.core.formatter.comment.insert_new_line_for_parameter"
     - possible values:   { INSERT, DO_NOT_INSERT }
     - default:           INSERT
```

Since:

3.1

See Also:

JavaCore.INSERT, JavaCore.DO_NOT_INSERT, Constant Field Values

FORMATTER_COMMENT_LINE_LENGTH

```
public static final String  FORMATTER_COMMENT_LINE_LENGTH
```

```
 FORMATTER / Option to specify the line length for comments.
     - option id:         "org.eclipse.jdt.core.formatter.comment.line_length"
     - possible values:   "<n>", where n is zero or a positive integer
     - default:           "80"
```

Since:

3.1

See Also:

Constant Field Values

FORMATTER_COMMENT_COUNT_LINE_LENGTH_FROM_STARTING_POSITION

```
public static final String  FORMATTER_COMMENT_COUNT_LINE_LENGTH_FROM_STARTING_POSITION
```

```
 FORMATTER / Option to control whether comments' line length will be counted from their starting position
     - option id:         "org.eclipse.jdt.core.formatter.comment.count_line_length_from_starting_position"
     - possible values:   { TRUE, FALSE }
     - default:           TRUE
```

Since:

3.13

See Also:

Constant Field Values

FORMATTER_COMMENT_NEW_LINES_AT_BLOCK_BOUNDARIES

```
public static final String  FORMATTER_COMMENT_NEW_LINES_AT_BLOCK_BOUNDARIES
```

```
 FORMATTER / Option to control whether block comments will have new lines at boundaries
     - option id:         "org.eclipse.jdt.core.formatter.comment.new_lines_at_block_boundaries"
     - possible values:   { TRUE, FALSE }
     - default:           TRUE
```

Since:

3.6

See Also:

TRUE, FALSE, Constant Field Values

FORMATTER_COMMENT_NEW_LINES_AT_JAVADOC_BOUNDARIES

```
public static final String  FORMATTER_COMMENT_NEW_LINES_AT_JAVADOC_BOUNDARIES
```

```
 FORMATTER / Option to control whether javadoc comments will have new lines at boundaries
     - option id:         "org.eclipse.jdt.core.formatter.comment.new_lines_at_javadoc_boundaries"
     - possible values:   { TRUE, FALSE }
     - default:           TRUE
```

Since:

3.6

See Also:

TRUE, FALSE, Constant Field Values

FORMATTER_COMMENT_JAVADOC_DO_NOT_SEPARATE_BLOCK_TAGS

```
public static final String  FORMATTER_COMMENT_JAVADOC_DO_NOT_SEPARATE_BLOCK_TAGS
```

```
 FORMATTER / Option to control whether paragraph tags in javadoc comments are put with the content or on their own line
     - option id:         "org.eclipse.jdt.core.formatter.comment.javadoc_paragraphs_tags_with_content"
     - possible values:   { TRUE, FALSE }
     - default:           FALSE
```

Since:

3.34

See Also:

TRUE, FALSE, Constant Field Values

FORMATTER_COMPACT_ELSE_IF

```
public static final String  FORMATTER_COMPACT_ELSE_IF
```

```
 FORMATTER / Option to compact else/if
     - option id:         "org.eclipse.jdt.core.formatter.compact_else_if"
     - possible values:   { TRUE, FALSE }
     - default:           TRUE
```

Since:

3.0

See Also:

TRUE, FALSE, Constant Field Values

FORMATTER_CONTINUATION_INDENTATION

```
public static final String  FORMATTER_CONTINUATION_INDENTATION
```

```
 FORMATTER / Option to set the continuation indentation
     - option id:         "org.eclipse.jdt.core.formatter.continuation_indentation"
     - possible values:   "<n>", where n is zero or a positive integer
     - default:           "2"
```

Since:

3.0

See Also:

Constant Field Values

FORMATTER_CONTINUATION_INDENTATION_FOR_ARRAY_INITIALIZER

```
public static final String  FORMATTER_CONTINUATION_INDENTATION_FOR_ARRAY_INITIALIZER
```

```
 FORMATTER / Option to set the continuation indentation inside array initializer
     - option id:         "org.eclipse.jdt.core.formatter.continuation_indentation_for_array_initializer"
     - possible values:   "<n>", where n is zero or a positive integer
     - default:           "2"
```

Since:

3.0

See Also:

Constant Field Values

FORMATTER_USE_ON_OFF_TAGS

```
public static final String  FORMATTER_USE_ON_OFF_TAGS
```

```
 FORMATTER / Option to use the disabling and enabling tags defined respectively by the FORMATTER_DISABLING_TAG and the FORMATTER_ENABLING_TAG options.
     - option id:         "org.eclipse.jdt.core.formatter.use_on_off_tags"
     - possible values:   TRUE / FALSE
     - default:           TRUE
```

Since:

3.6

See Also:

Constant Field Values

FORMATTER_DISABLING_TAG

```
public static final String  FORMATTER_DISABLING_TAG
```

```
 FORMATTER / Option to define the tag to put in a comment to disable the formatting.
     - option id:         "org.eclipse.jdt.core.formatter.disabling_tag"
     - possible values:   String, with constraints mentioned below
     - default:           "@formatter:off"

 See the FORMATTER_ENABLING_TAG option to re-enable it.
```

Note that:

1. This tag is used by the formatter only if the `FORMATTER_USE_ON_OFF_TAGS` option is set to `TRUE`.
2. The tag name will be trimmed. Hence if it does contain white spaces at the beginning or at the end, they will not be taken into account while searching for the tag in the comments
3. If a tag is starting with a letter or digit, then it cannot be leaded by another letter or digit to be recognized (`"ToDisableFormatter"` will not be recognized as a disabling tag `"DisableFormatter"`, but `"To:DisableFormatter"` will be detected for either tag `"DisableFormatter"` or `":DisableFormatter"`).
   Respectively, a tag ending with a letter or digit cannot be followed by a letter or digit to be recognized (`"DisableFormatter1"` will not be recognized as a disabling tag `"DisableFormatter"`, but `"DisableFormatter:1"` will be detected either for tag `"DisableFormatter"` or `"DisableFormatter:"`)
4. As soon as the formatter encounters the defined disabling tag, it stops to format the code from the beginning of the comment including this tag. If it was already disabled, the tag has no special effect.

   For example, the second default enabling tag "**@formatter:off**" in the following snippet is not necessary as the formatter was already disabled since the first one:

   ```
   class X {
   // @formatter:off
   void foo1() {}
   // @formatter:off
   void foo2() {}
   void bar1() {}
   void bar2() {}
   }
   ```

5. If no enabling tag is found by the formatter after the disabling tag, then the end of the snippet won't be formatted.
   For example, when a disabling tag is put at the beginning of the code, then the entire content of a compilation unit is not formatted:

```
// @formatter:off
class X {
void foo1() {}
void foo2() {}
void bar1() {}
void bar2() {}
}
```

6. If a mix of disabling and enabling tags is done in the same comment, then the formatter will only take into account the last encountered tag in the comment.

   For example, in the following snippet, the formatter will be disabled after the comment:

```
class X {
/*
 * This is a comment with a mix of disabling and enabling tags:
 *  - @formatter:off
 *  - @formatter:on
 *  - @formatter:off
 * The formatter will stop to format from the beginning of this comment...
 */
void foo() {}
void bar() {}
}
```

7. The tag cannot include newline character (i.e. '\n') but it can have white spaces.
   E.g. "**format: off**" is a valid disabling tag.
   In the future, newlines may be used to support multiple disabling tags.
8. The tag can include line or block comments start/end tokens.

   If such tags are used, e.g. "**//J-**", then the single comment can also stop the formatting as shown in the following snippet:

```
//J-
// Formatting was stopped from comment above...
public class X {
//J+
// Formatting is restarted from here...
void foo() {}
```

   As any disabling tags, as soon as a comment includes it, the formatting stops from this comment:

```
public class X {
// Line comment including the disabling tag: //J-
// Formatting was stopped from comment above...
void   foo1()   {}
//J+
// Formatting restarts from here...
void   bar1()   {}
/*
 * Block comment including the disabling tag: //J+
 * The formatter stops from this comment...
 */
void   foo2()   {}
//J+
// Formatting restarts from here...
void   bar2()   {}
/**
 * Javadoc comment including the enabling tag: //J+
 * The formatter stops from this comment...
 */
void   foo3()   {}
}
```

Since:

3.6

See Also:

Constant Field Values

FORMATTER_ENABLING_TAG

```
public static final String  FORMATTER_ENABLING_TAG
```

```
 FORMATTER / Option to define the tag to put in a comment to re-enable the formatting after it has been disabled (see FORMATTER_DISABLING_TAG)
     - option id:         "org.eclipse.jdt.core.formatter.enabling_tag"
     - possible values:   String, with constraints mentioned below
     - default:           "@formatter:on"
```

Note that:

1. This tag is used by the formatter only if the `FORMATTER_USE_ON_OFF_TAGS` option is set to `TRUE`.
2. The tag name will be trimmed. Hence if it does contain white spaces at the beginning or at the end, they will not be taken into account while searching for the tag in the comments
3. If a tag is starting with a letter or digit, then it cannot be leaded by another letter or digit to be recognized (`"ReEnableFormatter"` will not be recognized as an enabling tag `"EnableFormatter"`, but `"Re:EnableFormatter"` will be detected for either tag `"EnableFormatter"` or `":EnableFormatter"`).
   Respectively, a tag ending with a letter or digit cannot be followed by a letter or digit to be recognized (`"EnableFormatter1"` will not be recognized as an enabling tag `"EnableFormatter"`, but `"EnableFormatter:1"` will be detected either for tag `"EnableFormatter"` or `"EnableFormatter:"`)
4. As soon as the formatter encounters the defined enabling tag, it re-starts to format the code just after the comment including this tag. If it was already active, i.e. already re-enabled or never disabled, the tag has no special effect.

   For example, the default enabling tag "**@formatter:on**" in the following snippet is not necessary as the formatter has never been disabled:

   ```
   class X {
   void foo1() {}
   void foo2() {}
   // @formatter:on
   void bar1() {}
   void bar2() {}
   }
   ```

   Or, in the following other snippet, the second enabling tag is not necessary as the formatting will have been re-enabled by the first one:

   ```
   class X {
   // @formatter:off
   void foo1() {}
   void foo2() {}
   // @formatter:on
   void bar1() {}
   // @formatter:on
   void bar2() {}
   }
   ```

5. If a mix of disabling and enabling tags is done in the same comment, then the formatter will only take into account the last encountered tag in the comment.

   For example, in the following snippet, the formatter will be re-enabled after the comment:

   ```
   // @formatter:off
   class X {
   /*
    * This is a comment with a mix of disabling and enabling tags:
    *  - @formatter:on
    *  - @formatter:off
    *  - @formatter:on
    * The formatter will restart to format after this comment...
    */
   void foo() {}
   void bar() {}
   }
   ```

6. The tag cannot include newline character (i.e. '\n') but it can have white spaces.
   E.g. "**format: on**" is a valid enabling tag
   In the future, newlines may be used to support multiple enabling tags.
7. The tag can include line or block comments start/end tokens. Javadoc tokens are not considered as valid tags.

   If such tags are used, e.g. "**//J+**", then the single comment can also start the formatting as shown in the following snippet:

   ```
   //J-
   // Formatting was stopped from comment above...
   public class X {
   //J+
   // Formatting restarts from here...
   void foo() {}
   }
   ```

   As any enabling tags, as soon as a comment includes it, the formatting restarts just after the comment:

   ```
   public class X {
   //J-
   // Formatting was stopped from comment above...
   void   foo1()   {}
   // Line comment including the enabling tag: //J+
   // Formatting restarts from here...
   void   bar1()   {}
   //J-
   // Formatting was stopped from comment above...
   void   foo2()   {}
   /*
    * Block comment including the enabling tag: //J+
    * The formatter restarts after this comment...
   ```

```
 */
// Formatting restarts from here...
void   bar2()   {}
//J-
// Formatting was stopped from comment above...
void   foo3()   {}
/**
 * Javadoc comment including the enabling tag: //J+
 * The formatter restarts after this comment...
 */
void   bar3()   {}
}
```

Since:

3.6

See Also:

Constant Field Values

FORMATTER_INDENT_BODY_DECLARATIONS_COMPARE_TO_ANNOTATION_DECLARATION_HEADER

```
public static final String  FORMATTER_INDENT_BODY_DECLARATIONS_COMPARE_TO_ANNOTATION_DECLARATION_HEADER
```

```
 FORMATTER / Option to indent body declarations compare to its enclosing annotation declaration header
     - option id:         "org.eclipse.jdt.core.formatter.indent_body_declarations_compare_to_annotation_declaration_header"
     - possible values:   { TRUE, FALSE }
     - default:           TRUE
```

Since:

3.2

See Also:

TRUE, FALSE, Constant Field Values

FORMATTER_INDENT_BODY_DECLARATIONS_COMPARE_TO_ENUM_CONSTANT_HEADER

```
public static final String  FORMATTER_INDENT_BODY_DECLARATIONS_COMPARE_TO_ENUM_CONSTANT_HEADER
```

```
 FORMATTER / Option to indent body declarations compare to its enclosing enum constant header
     - option id:         "org.eclipse.jdt.core.formatter.indent_body_declarations_compare_to_enum_constant_header"
     - possible values:   { TRUE, FALSE }
     - default:           TRUE
```

Since:

3.1

See Also:

TRUE, FALSE, Constant Field Values

FORMATTER_INDENT_BODY_DECLARATIONS_COMPARE_TO_ENUM_DECLARATION_HEADER

```
public static final String  FORMATTER_INDENT_BODY_DECLARATIONS_COMPARE_TO_ENUM_DECLARATION_HEADER
```

```
 FORMATTER / Option to indent body declarations compare to its enclosing enum declaration header
     - option id:         "org.eclipse.jdt.core.formatter.indent_body_declarations_compare_to_enum_declaration_header"
     - possible values:   { TRUE, FALSE }
     - default:           TRUE
```

Since:

3.1

See Also:

TRUE, FALSE, Constant Field Values

FORMATTER_INDENT_BODY_DECLARATIONS_COMPARE_TO_RECORD_HEADER

```
public static final String  FORMATTER_INDENT_BODY_DECLARATIONS_COMPARE_TO_RECORD_HEADER
```

```
 FORMATTER / Option to indent body declarations compare to its enclosing record header
     - option id:         "org.eclipse.jdt.core.formatter.indent_body_declarations_compare_to_record_header"
     - possible values:   { TRUE, FALSE }
     - default:           TRUE
```

Since:

3.22

See Also:

TRUE, FALSE, Constant Field Values

FORMATTER_INDENT_BODY_DECLARATIONS_COMPARE_TO_TYPE_HEADER

```
public static final String  FORMATTER_INDENT_BODY_DECLARATIONS_COMPARE_TO_TYPE_HEADER
```

```
 FORMATTER / Option to indent body declarations compare to its enclosing type header
     - option id:         "org.eclipse.jdt.core.formatter.indent_body_declarations_compare_to_type_header"
     - possible values:   { TRUE, FALSE }
     - default:           TRUE
```

Since:

3.0

See Also:

TRUE, FALSE, Constant Field Values

FORMATTER_INDENT_BREAKS_COMPARE_TO_CASES

```
public static final String  FORMATTER_INDENT_BREAKS_COMPARE_TO_CASES
```

```
 FORMATTER / Option to indent breaks compare to cases
     - option id:         "org.eclipse.jdt.core.formatter.indent_breaks_compare_to_cases"
     - possible values:   { TRUE, FALSE }
     - default:           TRUE
```

Since:

3.0

See Also:

TRUE, FALSE, Constant Field Values

FORMATTER_INDENT_EMPTY_LINES

```
public static final String  FORMATTER_INDENT_EMPTY_LINES
```

```
 FORMATTER / Option to indent empty lines
     - option id:         "org.eclipse.jdt.core.formatter.indent_empty_lines"
     - possible values:   { TRUE, FALSE }
     - default:           FALSE
```

Since:

3.2

See Also:

TRUE, FALSE, Constant Field Values

FORMATTER_INDENT_STATEMENTS_COMPARE_TO_BLOCK

```
public static final String  FORMATTER_INDENT_STATEMENTS_COMPARE_TO_BLOCK
```

```
 FORMATTER / Option to indent statements inside a block
     - option id:         "org.eclipse.jdt.core.formatter.indent_statements_compare_to_block"
     - possible values:   { TRUE, FALSE }
     - default:           TRUE
```

Since:

3.0

See Also:

TRUE, FALSE, Constant Field Values

FORMATTER_INDENT_STATEMENTS_COMPARE_TO_BODY

```
public static final String  FORMATTER_INDENT_STATEMENTS_COMPARE_TO_BODY
```

```
 FORMATTER / Option to indent statements inside the body of a method or a constructor
     - option id:         "org.eclipse.jdt.core.formatter.indent_statements_compare_to_body"
     - possible values:   { TRUE, FALSE }
     - default:           TRUE
```

Since:

3.0

See Also:

TRUE, FALSE, Constant Field Values

FORMATTER_INDENT_SWITCHSTATEMENTS_COMPARE_TO_CASES

```
public static final String  FORMATTER_INDENT_SWITCHSTATEMENTS_COMPARE_TO_CASES
```

```
 FORMATTER / Option to indent switch statements compare to cases
     - option id:         "org.eclipse.jdt.core.formatter.indent_switchstatements_compare_to_cases"
     - possible values:   { TRUE, FALSE }
     - default:           TRUE
```

Since:

3.0

See Also:

TRUE, FALSE, Constant Field Values

FORMATTER_INDENT_SWITCHSTATEMENTS_COMPARE_TO_SWITCH

```
public static final String  FORMATTER_INDENT_SWITCHSTATEMENTS_COMPARE_TO_SWITCH
```

```
 FORMATTER / Option to indent switch statements compare to switch
     - option id:         "org.eclipse.jdt.core.formatter.indent_switchstatements_compare_to_switch"
     - possible values:   { TRUE, FALSE }
     - default:           TRUE
```

Since:

3.0

See Also:

TRUE, FALSE, Constant Field Values

FORMATTER_INDENTATION_SIZE

```
public static final String  FORMATTER_INDENTATION_SIZE
```

```
 FORMATTER / Option to specify the equivalent number of spaces that represents one indentation
     - option id:         "org.eclipse.jdt.core.formatter.indentation.size"
     - possible values:   "<n>", where n is zero or a positive integer
     - default:           "4"
```

This option is used only if the tab char is set to MIXED.

Since:

3.1

See Also:

FORMATTER_TAB_CHAR, Constant Field Values

FORMATTER_TEXT_BLOCK_INDENTATION

```
public static final String  FORMATTER_TEXT_BLOCK_INDENTATION
```

```
 FORMATTER / Option to specify how text blocks are indented
     - option id:         "org.eclipse.jdt.core.formatter.text_block_indentation"
     - possible values:   { INDENT_PRESERVE, INDENT_BY_ONE, INDENT_DEFAULT, INDENT_ON_COLUMN }
     - default:           INDENT_DEFAULT
```

Since:

3.20

See Also:

INDENT_PRESERVE, INDENT_BY_ONE, INDENT_DEFAULT, INDENT_ON_COLUMN, Constant Field Values

FORMATTER_INSERT_NEW_LINE_AFTER_ANNOTATION

```
@Deprecated
public static final String  FORMATTER_INSERT_NEW_LINE_AFTER_ANNOTATION
```

Deprecated.
All new options must be enabled to activate old strategy FORMATTER_INSERT_NEW_LINE_AFTER_ANNOTATION_ON_MEMBER FORMATTER_INSERT_NEW_LINE_AFTER_ANNOTATION_ON_LOCAL_VARIABLE FORMATTER_INSERT_NEW_LINE_AFTER_ANNOTATION_ON_PARAMETER

```
FORMATTER / Option to insert a new line after an annotation
     - option id:         "org.eclipse.jdt.core.formatter.insert_new_line_after_annotation"
     - possible values:   { INSERT, DO_NOT_INSERT }
     - default:           INSERT
```

Since:

3.1

See Also:

JavaCore.INSERT, JavaCore.DO_NOT_INSERT, Constant Field Values

FORMATTER_INSERT_NEW_LINE_AFTER_ANNOTATION_ON_MEMBER

```
@Deprecated
public static final String  FORMATTER_INSERT_NEW_LINE_AFTER_ANNOTATION_ON_MEMBER
```

Deprecated.
All new options must be enabled to activate old strategy FORMATTER_INSERT_NEW_LINE_AFTER_ANNOTATION_ON_FIELD FORMATTER_INSERT_NEW_LINE_AFTER_ANNOTATION_ON_METHOD FORMATTER_INSERT_NEW_LINE_AFTER_ANNOTATION_ON_PACKAGE FORMATTER_INSERT_NEW_LINE_AFTER_ANNOTATION_ON_TYPE

```
FORMATTER / Option to insert a new line after an annotation on a member (package, class, method, field declaration)
     - option id:         "org.eclipse.jdt.core.formatter.insert_new_line_after_annotation_on_member"
     - possible values:   { INSERT, DO_NOT_INSERT }
     - default:           INSERT
```

Since:

3.4

See Also:

JavaCore.INSERT, JavaCore.DO_NOT_INSERT, Constant Field Values

FORMATTER_INSERT_NEW_LINE_AFTER_ANNOTATION_ON_ENUM_CONSTANT

```
public static final String  FORMATTER_INSERT_NEW_LINE_AFTER_ANNOTATION_ON_ENUM_CONSTANT
```

```
FORMATTER / Option to insert a new line after an annotation on an enum constant declaration
     - option id:         "org.eclipse.jdt.core.formatter.insert_new_line_after_annotation_on_enum_constant"
     - possible values:   { INSERT, DO_NOT_INSERT }
     - default:           INSERT
```

Since:

3.12

See Also:

JavaCore.INSERT, JavaCore.DO_NOT_INSERT, Constant Field Values

FORMATTER_INSERT_NEW_LINE_AFTER_ANNOTATION_ON_FIELD

```
public static final String  FORMATTER_INSERT_NEW_LINE_AFTER_ANNOTATION_ON_FIELD
```

```
FORMATTER / Option to insert a new line after an annotation on a field declaration
     - option id:         "org.eclipse.jdt.core.formatter.insert_new_line_after_annotation_on_field"
     - possible values:   { INSERT, DO_NOT_INSERT }
     - default:           INSERT
```

Since:

3.7

See Also:

JavaCore.INSERT, JavaCore.DO_NOT_INSERT, Constant Field Values

FORMATTER_INSERT_NEW_LINE_AFTER_ANNOTATION_ON_METHOD

```
public static final String  FORMATTER_INSERT_NEW_LINE_AFTER_ANNOTATION_ON_METHOD
```

```
FORMATTER / Option to insert a new line after an annotation on a method declaration
     - option id:         "org.eclipse.jdt.core.formatter.insert_new_line_after_annotation_on_method"
     - possible values:   { INSERT, DO_NOT_INSERT }
     - default:           INSERT
```

Since:

3.7

See Also:

JavaCore.INSERT, JavaCore.DO_NOT_INSERT, Constant Field Values

FORMATTER_INSERT_NEW_LINE_AFTER_ANNOTATION_ON_PACKAGE

```
public static final String  FORMATTER_INSERT_NEW_LINE_AFTER_ANNOTATION_ON_PACKAGE
```

```
 FORMATTER / Option to insert a new line after an annotation on a package declaration
     - option id:         "org.eclipse.jdt.core.formatter.insert_new_line_after_annotation_on_package"
     - possible values:   { INSERT, DO_NOT_INSERT }
     - default:           INSERT
```

Since:

3.7

See Also:

JavaCore.INSERT, JavaCore.DO_NOT_INSERT, Constant Field Values

FORMATTER_INSERT_NEW_LINE_AFTER_ANNOTATION_ON_TYPE

```
public static final String  FORMATTER_INSERT_NEW_LINE_AFTER_ANNOTATION_ON_TYPE
```

```
 FORMATTER / Option to insert a new line after an annotation on a type declaration
     - option id:         "org.eclipse.jdt.core.formatter.insert_new_line_after_annotation_on_type"
     - possible values:   { INSERT, DO_NOT_INSERT }
     - default:           INSERT
```

Since:

3.7

See Also:

JavaCore.INSERT, JavaCore.DO_NOT_INSERT, Constant Field Values

FORMATTER_INSERT_NEW_LINE_AFTER_TYPE_ANNOTATION

```
public static final String  FORMATTER_INSERT_NEW_LINE_AFTER_TYPE_ANNOTATION
```

```
 FORMATTER / Option to insert a new line after a type annotation
     - option id:         "org.eclipse.jdt.core.formatter.insert_new_line_after_type_annotation"
     - possible values:   { INSERT, DO_NOT_INSERT }
     - default:           DO_NOT_INSERT
```

Since:

3.10

See Also:

JavaCore.INSERT, JavaCore.DO_NOT_INSERT, Constant Field Values

FORMATTER_INSERT_NEW_LINE_AFTER_ANNOTATION_ON_PARAMETER

```
public static final String  FORMATTER_INSERT_NEW_LINE_AFTER_ANNOTATION_ON_PARAMETER
```

```
 FORMATTER / Option to insert a new line after an annotation on a parameter
     - option id:         "org.eclipse.jdt.core.formatter.insert_new_line_after_annotation_on_parameter"
     - possible values:   { INSERT, DO_NOT_INSERT }
     - default:           DO_NOT_INSERT
```

Since:

3.4

See Also:

JavaCore.INSERT, JavaCore.DO_NOT_INSERT, Constant Field Values

FORMATTER_INSERT_NEW_LINE_AFTER_ANNOTATION_ON_LOCAL_VARIABLE

```
public static final String  FORMATTER_INSERT_NEW_LINE_AFTER_ANNOTATION_ON_LOCAL_VARIABLE
```

```
 FORMATTER / Option to insert a new line after an annotation on a local variable
     - option id:         "org.eclipse.jdt.core.formatter.insert_new_line_after_annotation_on_local_variable"
     - possible values:   { INSERT, DO_NOT_INSERT }
     - default:           INSERT
```

Since:

3.4

See Also:

JavaCore.INSERT, JavaCore.DO_NOT_INSERT, Constant Field Values

FORMATTER_INSERT_NEW_LINE_AFTER_LABEL

```
public static final String  FORMATTER_INSERT_NEW_LINE_AFTER_LABEL
```

```
 FORMATTER / Option to insert a new line after a label
     - option id:         "org.eclipse.jdt.core.formatter.insert_new_line_after_label"
     - possible values:   { INSERT, DO_NOT_INSERT }
     - default:           DO_NOT_INSERT
```

Since:

3.6

See Also:

JavaCore.INSERT, JavaCore.DO_NOT_INSERT, Constant Field Values

FORMATTER_INSERT_NEW_LINE_AFTER_OPENING_BRACE_IN_ARRAY_INITIALIZER

```
public static final String  FORMATTER_INSERT_NEW_LINE_AFTER_OPENING_BRACE_IN_ARRAY_INITIALIZER
```

```
 FORMATTER / Option to insert a new line after the opening brace in an array initializer
     - option id:         "org.eclipse.jdt.core.formatter.insert_new_line_after_opening_brace_in_array_initializer"
     - possible values:   { INSERT, DO_NOT_INSERT }
     - default:           DO_NOT_INSERT
```

Since:

3.0

See Also:

JavaCore.INSERT, JavaCore.DO_NOT_INSERT, Constant Field Values

FORMATTER_INSERT_NEW_LINE_AT_END_OF_FILE_IF_MISSING

```
public static final String  FORMATTER_INSERT_NEW_LINE_AT_END_OF_FILE_IF_MISSING
```

```
 FORMATTER / Option to insert a new line at the end of the current file if missing
     - option id:         "org.eclipse.jdt.core.formatter.insert_new_line_at_end_of_file_if_missing"
     - possible values:   { INSERT, DO_NOT_INSERT }
     - default:           DO_NOT_INSERT
```

Since:

3.1

See Also:

JavaCore.INSERT, JavaCore.DO_NOT_INSERT, Constant Field Values

FORMATTER_INSERT_NEW_LINE_BEFORE_CATCH_IN_TRY_STATEMENT

```
public static final String  FORMATTER_INSERT_NEW_LINE_BEFORE_CATCH_IN_TRY_STATEMENT
```

```
 FORMATTER / Option to insert a new line before the catch keyword in try statement
     - option id:         "org.eclipse.jdt.core.formatter.insert_new_line_before_catch_in_try_statement"
     - possible values:   { INSERT, DO_NOT_INSERT }
     - default:           DO_NOT_INSERT
```

Since:

3.0

See Also:

JavaCore.INSERT, JavaCore.DO_NOT_INSERT, Constant Field Values

FORMATTER_INSERT_NEW_LINE_BEFORE_CLOSING_BRACE_IN_ARRAY_INITIALIZER

```
public static final String  FORMATTER_INSERT_NEW_LINE_BEFORE_CLOSING_BRACE_IN_ARRAY_INITIALIZER
```

```
 FORMATTER / Option to insert a new line before the closing brace in an array initializer
     - option id:         "org.eclipse.jdt.core.formatter.insert_new_line_before_closing_brace_in_array_initializer"
     - possible values:   { INSERT, DO_NOT_INSERT }
     - default:           DO_NOT_INSERT
```

Since:

3.0

See Also:

JavaCore.INSERT, JavaCore.DO_NOT_INSERT, Constant Field Values

FORMATTER_INSERT_NEW_LINE_BEFORE_ELSE_IN_IF_STATEMENT

```
public static final String  FORMATTER_INSERT_NEW_LINE_BEFORE_ELSE_IN_IF_STATEMENT
```

```
 FORMATTER / Option to insert a new line before the else keyword in if statement
     - option id:         "org.eclipse.jdt.core.formatter.insert_new_line_before_else_in_if_statement"
     - possible values:   { INSERT, DO_NOT_INSERT }
     - default:           DO_NOT_INSERT
```

Since:

3.0

See Also:

JavaCore.INSERT, JavaCore.DO_NOT_INSERT, Constant Field Values

FORMATTER_INSERT_NEW_LINE_BEFORE_FINALLY_IN_TRY_STATEMENT

```
public static final String  FORMATTER_INSERT_NEW_LINE_BEFORE_FINALLY_IN_TRY_STATEMENT
```

```
 FORMATTER / Option to insert a new line before the finally keyword in try statement
     - option id:         "org.eclipse.jdt.core.formatter.insert_new_line_before_finally_in_try_statement"
     - possible values:   { INSERT, DO_NOT_INSERT }
     - default:           DO_NOT_INSERT
```

Since:

3.0

See Also:

JavaCore.INSERT, JavaCore.DO_NOT_INSERT, Constant Field Values

FORMATTER_INSERT_NEW_LINE_BEFORE_WHILE_IN_DO_STATEMENT

```
public static final String  FORMATTER_INSERT_NEW_LINE_BEFORE_WHILE_IN_DO_STATEMENT
```

```
 FORMATTER / Option to insert a new line before while in do statement
     - option id:         "org.eclipse.jdt.core.formatter.insert_new_line_before_while_in_do_statement"
     - possible values:   { INSERT, DO_NOT_INSERT }
     - default:           DO_NOT_INSERT
```

Since:

3.0

See Also:

JavaCore.INSERT, JavaCore.DO_NOT_INSERT, Constant Field Values

FORMATTER_INSERT_NEW_LINE_IN_EMPTY_ANNOTATION_DECLARATION

```
@Deprecated
public static final String  FORMATTER_INSERT_NEW_LINE_IN_EMPTY_ANNOTATION_DECLARATION
```

Deprecated.
Use `FORMATTER_KEEP_ANNOTATION_DECLARATION_ON_ONE_LINE` instead.

```
 FORMATTER / Option to insert a new line in an empty annotation declaration
     - option id:         "org.eclipse.jdt.core.formatter.insert_new_line_in_empty_annotation_declaration"
     - possible values:   { INSERT, DO_NOT_INSERT }
     - default:           INSERT
```

Since:

3.2

See Also:

JavaCore.INSERT, JavaCore.DO_NOT_INSERT, Constant Field Values

FORMATTER_INSERT_NEW_LINE_IN_EMPTY_ANONYMOUS_TYPE_DECLARATION

```
@Deprecated
public static final String  FORMATTER_INSERT_NEW_LINE_IN_EMPTY_ANONYMOUS_TYPE_DECLARATION
```

Deprecated.
Use `FORMATTER_KEEP_ANONYMOUS_TYPE_DECLARATION_ON_ONE_LINE` instead.

```
FORMATTER / Option to insert a new line in an empty anonymous type declaration
    - option id:         "org.eclipse.jdt.core.formatter.insert_new_line_in_empty_anonymous_type_declaration"
    - possible values:   { INSERT, DO_NOT_INSERT }
    - default:           INSERT
```

Since:

3.0

See Also:

`JavaCore.INSERT`, `JavaCore.DO_NOT_INSERT`, Constant Field Values

FORMATTER_INSERT_NEW_LINE_IN_EMPTY_BLOCK

```
@Deprecated
public static final String  FORMATTER_INSERT_NEW_LINE_IN_EMPTY_BLOCK
```

Deprecated.
Use `FORMATTER_KEEP_IF_THEN_BODY_BLOCK_ON_ONE_LINE`, `FORMATTER_KEEP_LOOP_BODY_BLOCK_ON_ONE_LINE`, `FORMATTER_KEEP_CODE_BLOCK_ON_ONE_LINE`, and `FORMATTER_KEEP_LAMBDA_BODY_BLOCK_ON_ONE_LINE` instead.

```
FORMATTER / Option to insert a new line in an empty block
    - option id:         "org.eclipse.jdt.core.formatter.insert_new_line_in_empty_block"
    - possible values:   { INSERT, DO_NOT_INSERT }
    - default:           INSERT
```

Since:

3.0

See Also:

`JavaCore.INSERT`, `JavaCore.DO_NOT_INSERT`, Constant Field Values

FORMATTER_INSERT_NEW_LINE_IN_EMPTY_ENUM_CONSTANT

```
@Deprecated
public static final String  FORMATTER_INSERT_NEW_LINE_IN_EMPTY_ENUM_CONSTANT
```

Deprecated.
Use `FORMATTER_KEEP_ENUM_CONSTANT_DECLARATION_ON_ONE_LINE` instead.

```
FORMATTER / Option to insert a new line in an empty enum constant
    - option id:         "org.eclipse.jdt.core.formatter.insert_new_line_in_empty_enum_constant"
    - possible values:   { INSERT, DO_NOT_INSERT }
    - default:           INSERT
```

Since:

3.1

See Also:

`JavaCore.INSERT`, `JavaCore.DO_NOT_INSERT`, Constant Field Values

FORMATTER_INSERT_NEW_LINE_IN_EMPTY_ENUM_DECLARATION

```
@Deprecated
public static final String  FORMATTER_INSERT_NEW_LINE_IN_EMPTY_ENUM_DECLARATION
```

Deprecated.
Use `FORMATTER_KEEP_ENUM_DECLARATION_ON_ONE_LINE` instead.

```
FORMATTER / Option to insert a new line in an empty enum declaration
    - option id:         "org.eclipse.jdt.core.formatter.insert_new_line_in_empty_enum_declaration"
    - possible values:   { INSERT, DO_NOT_INSERT }
    - default:           INSERT
```

Since:

3.1

See Also:

`JavaCore.INSERT`, `JavaCore.DO_NOT_INSERT`, Constant Field Values

FORMATTER_INSERT_NEW_LINE_IN_EMPTY_METHOD_BODY

```
@Deprecated
public static final String  FORMATTER_INSERT_NEW_LINE_IN_EMPTY_METHOD_BODY
```

Deprecated.
Use `FORMATTER_KEEP_METHOD_BODY_ON_ONE_LINE` instead.

```
FORMATTER / Option to insert a new line in an empty method body
    - option id:         "org.eclipse.jdt.core.formatter.insert_new_line_in_empty_method_body"
    - possible values:   { INSERT, DO_NOT_INSERT }
    - default:           INSERT
```

Since:

3.0

See Also:

JavaCore.INSERT, JavaCore.DO_NOT_INSERT, Constant Field Values

FORMATTER_INSERT_NEW_LINE_IN_EMPTY_TYPE_DECLARATION

```
@Deprecated
public static final String  FORMATTER_INSERT_NEW_LINE_IN_EMPTY_TYPE_DECLARATION
```

Deprecated.
Use `FORMATTER_KEEP_TYPE_DECLARATION_ON_ONE_LINE` instead.

```
FORMATTER / Option to insert a new line in an empty type declaration
    - option id:         "org.eclipse.jdt.core.formatter.insert_new_line_in_empty_type_declaration"
    - possible values:   { INSERT, DO_NOT_INSERT }
    - default:           INSERT
```

Since:

3.0

See Also:

JavaCore.INSERT, JavaCore.DO_NOT_INSERT, Constant Field Values

FORMATTER_INSERT_SPACE_AFTER_AND_IN_TYPE_PARAMETER

```
public static final String  FORMATTER_INSERT_SPACE_AFTER_AND_IN_TYPE_PARAMETER
```

```
FORMATTER / Option to insert a space after and in wilcard
    - option id:         "org.eclipse.jdt.core.formatter.insert_space_after_and_in_type_parameter"
    - possible values:   { INSERT, DO_NOT_INSERT }
    - default:           INSERT
```

Since:

3.1

See Also:

JavaCore.INSERT, JavaCore.DO_NOT_INSERT, Constant Field Values

FORMATTER_INSERT_SPACE_AFTER_ARROW_IN_SWITCH_CASE

```
public static final String  FORMATTER_INSERT_SPACE_AFTER_ARROW_IN_SWITCH_CASE
```

```
FORMATTER / Option to insert a space after arrow in switch case
    - option id:         "org.eclipse.jdt.core.formatter.insert_space_after_arrow_in_switch_case"
    - possible values:   { INSERT, DO_NOT_INSERT }
    - default:           INSERT
```

Since:

3.18

See Also:

JavaCore.INSERT, JavaCore.DO_NOT_INSERT, Constant Field Values

FORMATTER_INSERT_SPACE_AFTER_ARROW_IN_SWITCH_DEFAULT

```
public static final String  FORMATTER_INSERT_SPACE_AFTER_ARROW_IN_SWITCH_DEFAULT
```

```
FORMATTER / Option to insert a space after arrow in switch default
```

```
     - option id:         "org.eclipse.jdt.core.formatter.insert_space_after_arrow_in_switch_default"
     - possible values:   { INSERT, DO_NOT_INSERT }
     - default:           INSERT
```

Since:

3.18

See Also:

JavaCore.INSERT, JavaCore.DO_NOT_INSERT, Constant Field Values

FORMATTER_INSERT_SPACE_AFTER_ASSIGNMENT_OPERATOR

```
public static final String  FORMATTER_INSERT_SPACE_AFTER_ASSIGNMENT_OPERATOR
```

```
 FORMATTER / Option to insert a space after an assignment operator
     - option id:         "org.eclipse.jdt.core.formatter.insert_space_after_assignment_operator"
     - possible values:   { INSERT, DO_NOT_INSERT }
     - default:           INSERT
```

Since:

3.0

See Also:

JavaCore.INSERT, JavaCore.DO_NOT_INSERT, Constant Field Values

FORMATTER_INSERT_SPACE_AFTER_AT_IN_ANNOTATION

```
public static final String  FORMATTER_INSERT_SPACE_AFTER_AT_IN_ANNOTATION
```

```
 FORMATTER / Option to insert a space after at in annotation
     - option id:         "org.eclipse.jdt.core.formatter.insert_space_after_at_in_annotation"
     - possible values:   { INSERT, DO_NOT_INSERT }
     - default:           INSERT
```

Since:

3.1

See Also:

JavaCore.INSERT, JavaCore.DO_NOT_INSERT, Constant Field Values

FORMATTER_INSERT_SPACE_AFTER_AT_IN_ANNOTATION_TYPE_DECLARATION

```
public static final String  FORMATTER_INSERT_SPACE_AFTER_AT_IN_ANNOTATION_TYPE_DECLARATION
```

```
 FORMATTER / Option to insert a space after at in annotation type declaration
     - option id:         "org.eclipse.jdt.core.formatter.insert_space_after_at_in_annotation_type_declaration"
     - possible values:   { INSERT, DO_NOT_INSERT }
     - default:           DO_NOT_INSERT
```

Since:

3.1

See Also:

JavaCore.INSERT, JavaCore.DO_NOT_INSERT, Constant Field Values

FORMATTER_INSERT_SPACE_AFTER_BINARY_OPERATOR

```
public static final String  FORMATTER_INSERT_SPACE_AFTER_BINARY_OPERATOR
```

Deprecated.
Use the new settings instead: FORMATTER_INSERT_SPACE_AFTER_MULTIPLICATIVE_OPERATOR, FORMATTER_INSERT_SPACE_AFTER_ADDITIVE_OPERATOR, FORMATTER_INSERT_SPACE_AFTER_STRING_CONCATENATION, FORMATTER_INSERT_SPACE_AFTER_SHIFT_OPERATOR, FORMATTER_INSERT_SPACE_AFTER_RELATIONAL_OPERATOR, FORMATTER_INSERT_SPACE_AFTER_BITWISE_OPERATOR, FORMATTER_INSERT_SPACE_AFTER_LOGICAL_OPERATOR

```
 FORMATTER / Option to insert a space after a binary operator
     - option id:         "org.eclipse.jdt.core.formatter.insert_space_after_binary_operator"
     - possible values:   { INSERT, DO_NOT_INSERT }
     - default:           INSERT
```

Since:

3.0

See Also:

JavaCore.INSERT, JavaCore.DO_NOT_INSERT, Constant Field Values

FORMATTER_INSERT_SPACE_AFTER_MULTIPLICATIVE_OPERATOR

```
public static final String  FORMATTER_INSERT_SPACE_AFTER_MULTIPLICATIVE_OPERATOR
```

```
 FORMATTER / Option to insert a space after a multiplicative operator (*, /, %)
     - option id:         "org.eclipse.jdt.core.formatter.insert_space_after_multiplicative_operator"
     - possible values:   { INSERT, DO_NOT_INSERT }
     - default:           INSERT
```

Since:

3.17

See Also:

JavaCore.INSERT, JavaCore.DO_NOT_INSERT, Constant Field Values

FORMATTER_INSERT_SPACE_AFTER_ADDITIVE_OPERATOR

```
public static final String  FORMATTER_INSERT_SPACE_AFTER_ADDITIVE_OPERATOR
```

```
 FORMATTER / Option to insert a space after an additive operator (+, -)
     - option id:         "org.eclipse.jdt.core.formatter.insert_space_after_additive_operator"
     - possible values:   { INSERT, DO_NOT_INSERT }
     - default:           INSERT
```

Since:

3.17

See Also:

JavaCore.INSERT, JavaCore.DO_NOT_INSERT, Constant Field Values

FORMATTER_INSERT_SPACE_AFTER_STRING_CONCATENATION

```
public static final String  FORMATTER_INSERT_SPACE_AFTER_STRING_CONCATENATION
```

```
 FORMATTER / Option to insert a space after a string concatenation operator
     - option id:         "org.eclipse.jdt.core.formatter.insert_space_after_string_concatenation"
     - possible values:   { INSERT, DO_NOT_INSERT }
     - default:           INSERT
```

Since:

3.17

See Also:

JavaCore.INSERT, JavaCore.DO_NOT_INSERT, Constant Field Values

FORMATTER_INSERT_SPACE_AFTER_SHIFT_OPERATOR

```
public static final String  FORMATTER_INSERT_SPACE_AFTER_SHIFT_OPERATOR
```

```
 FORMATTER / Option to insert a space after a shift operator (<<, >>, >>>)
     - option id:         "org.eclipse.jdt.core.formatter.insert_space_after_shift_operator"
     - possible values:   { INSERT, DO_NOT_INSERT }
     - default:           INSERT
```

Since:

3.17

See Also:

JavaCore.INSERT, JavaCore.DO_NOT_INSERT, Constant Field Values

FORMATTER_INSERT_SPACE_AFTER_RELATIONAL_OPERATOR

```
public static final String  FORMATTER_INSERT_SPACE_AFTER_RELATIONAL_OPERATOR
```

```
 FORMATTER / Option to insert a space after a relational operator (<, >, <=, >=, ==, !=)
     - option id:         "org.eclipse.jdt.core.formatter.insert_space_after_relational_operator"
     - possible values:   { INSERT, DO_NOT_INSERT }
     - default:           INSERT
```

Since:

3.17

See Also:

JavaCore.INSERT, JavaCore.DO_NOT_INSERT, Constant Field Values

FORMATTER_INSERT_SPACE_AFTER_BITWISE_OPERATOR

```
public static final String  FORMATTER_INSERT_SPACE_AFTER_BITWISE_OPERATOR
```

```
 FORMATTER / Option to insert a space after a bitwise operator (&, ^, |)
     - option id:         "org.eclipse.jdt.core.formatter.insert_space_after_bitwise_operator"
     - possible values:   { INSERT, DO_NOT_INSERT }
     - default:           INSERT
```

Since:

3.17

See Also:

JavaCore.INSERT, JavaCore.DO_NOT_INSERT, Constant Field Values

FORMATTER_INSERT_SPACE_AFTER_LOGICAL_OPERATOR

```
public static final String  FORMATTER_INSERT_SPACE_AFTER_LOGICAL_OPERATOR
```

```
 FORMATTER / Option to insert a space after a logical operator (&&, ||)
     - option id:         "org.eclipse.jdt.core.formatter.insert_space_after_logical_operator"
     - possible values:   { INSERT, DO_NOT_INSERT }
     - default:           INSERT
```

Since:

3.17

See Also:

JavaCore.INSERT, JavaCore.DO_NOT_INSERT, Constant Field Values

FORMATTER_INSERT_SPACE_AFTER_CLOSING_ANGLE_BRACKET_IN_TYPE_ARGUMENTS

```
public static final String  FORMATTER_INSERT_SPACE_AFTER_CLOSING_ANGLE_BRACKET_IN_TYPE_ARGUMENTS
```

```
 FORMATTER / Option to insert a space after the closing angle bracket in explicit type arguments on method/constructor invocations
     - option id:         "org.eclipse.jdt.core.formatter.insert_space_after_closing_angle_bracket_in_type_arguments"
     - possible values:   { INSERT, DO_NOT_INSERT }
     - default:           DO_NOT_INSERT
```

Since:

3.1

See Also:

JavaCore.INSERT, JavaCore.DO_NOT_INSERT, Constant Field Values

FORMATTER_INSERT_SPACE_AFTER_CLOSING_ANGLE_BRACKET_IN_TYPE_PARAMETERS

```
public static final String  FORMATTER_INSERT_SPACE_AFTER_CLOSING_ANGLE_BRACKET_IN_TYPE_PARAMETERS
```

```
 FORMATTER / Option to insert a space after the closing angle bracket in type parameter declarations
     - option id:         "org.eclipse.jdt.core.formatter.insert_space_after_closing_angle_bracket_in_type_parameters"
     - possible values:   { INSERT, DO_NOT_INSERT }
     - default:           INSERT
```

Since:

3.1

See Also:

JavaCore.INSERT, JavaCore.DO_NOT_INSERT, Constant Field Values

FORMATTER_INSERT_SPACE_AFTER_CLOSING_BRACE_IN_BLOCK

```
public static final String  FORMATTER_INSERT_SPACE_AFTER_CLOSING_BRACE_IN_BLOCK
```

```
 FORMATTER / Option to insert a space after the closing brace of a block
     - option id:         "org.eclipse.jdt.core.formatter.insert_space_after_closing_brace_in_block"
     - possible values:   { INSERT, DO_NOT_INSERT }
     - default:           INSERT
```

Since:

3.0

See Also:

JavaCore.INSERT, JavaCore.DO_NOT_INSERT, Constant Field Values

FORMATTER_INSERT_SPACE_AFTER_CLOSING_PAREN_IN_CAST

```
public static final String  FORMATTER_INSERT_SPACE_AFTER_CLOSING_PAREN_IN_CAST
```

```
 FORMATTER / Option to insert a space after the closing parenthesis of a cast expression
     - option id:         "org.eclipse.jdt.core.formatter.insert_space_after_closing_paren_in_cast"
     - possible values:   { INSERT, DO_NOT_INSERT }
     - default:           INSERT
```

Since:

3.0

See Also:

JavaCore.INSERT, JavaCore.DO_NOT_INSERT, Constant Field Values

FORMATTER_INSERT_SPACE_AFTER_COLON_IN_ASSERT

```
public static final String  FORMATTER_INSERT_SPACE_AFTER_COLON_IN_ASSERT
```

```
 FORMATTER / Option to insert a space after the colon in an assert statement
     - option id:         "org.eclipse.jdt.core.formatter.insert_space_after_colon_in_assert"
     - possible values:   { INSERT, DO_NOT_INSERT }
     - default:           INSERT
```

Since:

3.0

See Also:

JavaCore.INSERT, JavaCore.DO_NOT_INSERT, Constant Field Values

FORMATTER_INSERT_SPACE_AFTER_COLON_IN_CASE

```
public static final String  FORMATTER_INSERT_SPACE_AFTER_COLON_IN_CASE
```

```
 FORMATTER / Option to insert a space after colon in a case statement when a opening brace follows the colon
     - option id:         "org.eclipse.jdt.core.formatter.insert_space_after_colon_in_case"
     - possible values:   { INSERT, DO_NOT_INSERT }
     - default:           INSERT
```

Since:

3.0

See Also:

JavaCore.INSERT, JavaCore.DO_NOT_INSERT, Constant Field Values

FORMATTER_INSERT_SPACE_AFTER_COLON_IN_CONDITIONAL

```
public static final String  FORMATTER_INSERT_SPACE_AFTER_COLON_IN_CONDITIONAL
```

```
 FORMATTER / Option to insert a space after the colon in a conditional expression
     - option id:         "org.eclipse.jdt.core.formatter.insert_space_after_colon_in_conditional"
     - possible values:   { INSERT, DO_NOT_INSERT }
     - default:           INSERT
```

Since:

3.0

See Also:

JavaCore.INSERT, JavaCore.DO_NOT_INSERT, Constant Field Values

FORMATTER_INSERT_SPACE_AFTER_COLON_IN_FOR

```
public static final String  FORMATTER_INSERT_SPACE_AFTER_COLON_IN_FOR
```

```
 FORMATTER / Option to insert a space after colon in a for statement
     - option id:         "org.eclipse.jdt.core.formatter.insert_space_after_colon_in_for"
     - possible values:   { INSERT, DO_NOT_INSERT }
     - default:           INSERT
```

Since:

3.1

See Also:

JavaCore.INSERT, JavaCore.DO_NOT_INSERT, Constant Field Values

FORMATTER_INSERT_SPACE_AFTER_COLON_IN_LABELED_STATEMENT

```
public static final String  FORMATTER_INSERT_SPACE_AFTER_COLON_IN_LABELED_STATEMENT
```

```
 FORMATTER / Option to insert a space after the colon in a labeled statement
     - option id:         "org.eclipse.jdt.core.formatter.insert_space_after_colon_in_labeled_statement"
     - possible values:   { INSERT, DO_NOT_INSERT }
     - default:           INSERT
```

Since:

3.0

See Also:

JavaCore.INSERT, JavaCore.DO_NOT_INSERT, Constant Field Values

FORMATTER_INSERT_SPACE_AFTER_COMMA_IN_ALLOCATION_EXPRESSION

```
public static final String  FORMATTER_INSERT_SPACE_AFTER_COMMA_IN_ALLOCATION_EXPRESSION
```

```
 FORMATTER / Option to insert a space after the comma in an allocation expression
     - option id:         "org.eclipse.jdt.core.formatter.insert_space_after_comma_in_allocation_expression"
     - possible values:   { INSERT, DO_NOT_INSERT }
     - default:           INSERT
```

Since:

3.0

See Also:

JavaCore.INSERT, JavaCore.DO_NOT_INSERT, Constant Field Values

FORMATTER_INSERT_SPACE_AFTER_COMMA_IN_ANNOTATION

```
public static final String  FORMATTER_INSERT_SPACE_AFTER_COMMA_IN_ANNOTATION
```

```
 FORMATTER / Option to insert a space after the comma in annotation
     - option id:         "org.eclipse.jdt.core.formatter.insert_space_after_comma_in_annotation"
     - possible values:   { INSERT, DO_NOT_INSERT }
     - default:           INSERT
```

Since:

3.1

See Also:

JavaCore.INSERT, JavaCore.DO_NOT_INSERT, Constant Field Values

FORMATTER_INSERT_SPACE_AFTER_COMMA_IN_ARRAY_INITIALIZER

```
public static final String  FORMATTER_INSERT_SPACE_AFTER_COMMA_IN_ARRAY_INITIALIZER
```

```
 FORMATTER / Option to insert a space after the comma in an array initializer
     - option id:         "org.eclipse.jdt.core.formatter.insert_space_after_comma_in_array_initializer"
     - possible values:   { INSERT, DO_NOT_INSERT }
     - default:           INSERT
```

Since:

3.0

See Also:

JavaCore.INSERT, JavaCore.DO_NOT_INSERT, Constant Field Values

FORMATTER_INSERT_SPACE_AFTER_COMMA_IN_CONSTRUCTOR_DECLARATION_PARAMETERS

```
public static final String  FORMATTER_INSERT_SPACE_AFTER_COMMA_IN_CONSTRUCTOR_DECLARATION_PARAMETERS
```

```
 FORMATTER / Option to insert a space after the comma in the parameters of a constructor declaration
     - option id:         "org.eclipse.jdt.core.formatter.insert_space_after_comma_in_constructor_declaration_parameters"
     - possible values:   { INSERT, DO_NOT_INSERT }
     - default:           INSERT
```

Since:

3.0

See Also:

JavaCore.INSERT, JavaCore.DO_NOT_INSERT, Constant Field Values

FORMATTER_INSERT_SPACE_AFTER_COMMA_IN_CONSTRUCTOR_DECLARATION_THROWS

```
public static final String  FORMATTER_INSERT_SPACE_AFTER_COMMA_IN_CONSTRUCTOR_DECLARATION_THROWS
```

```
 FORMATTER / Option to insert a space after the comma in the exception names in a throws clause of a constructor declaration
     - option id:         "org.eclipse.jdt.core.formatter.insert_space_after_comma_in_constructor_declaration_throws"
     - possible values:   { INSERT, DO_NOT_INSERT }
     - default:           INSERT
```

Since:

3.0

See Also:

JavaCore.INSERT, JavaCore.DO_NOT_INSERT, Constant Field Values

FORMATTER_INSERT_SPACE_AFTER_COMMA_IN_ENUM_CONSTANT_ARGUMENTS

```
public static final String  FORMATTER_INSERT_SPACE_AFTER_COMMA_IN_ENUM_CONSTANT_ARGUMENTS
```

```
 FORMATTER / Option to insert a space after the comma in the arguments of an enum constant
     - option id:         "org.eclipse.jdt.core.formatter.insert_space_after_comma_in_enum_constant_arguments"
     - possible values:   { INSERT, DO_NOT_INSERT }
     - default:           INSERT
```

Since:

3.1

See Also:

JavaCore.INSERT, JavaCore.DO_NOT_INSERT, Constant Field Values

FORMATTER_INSERT_SPACE_AFTER_COMMA_IN_ENUM_DECLARATIONS

```
public static final String  FORMATTER_INSERT_SPACE_AFTER_COMMA_IN_ENUM_DECLARATIONS
```

```
 FORMATTER / Option to insert a space after the comma in enum declarations
     - option id:         "org.eclipse.jdt.core.formatter.insert_space_after_comma_in_enum_declarations"
     - possible values:   { INSERT, DO_NOT_INSERT }
     - default:           INSERT
```

Since:

3.1

See Also:

JavaCore.INSERT, JavaCore.DO_NOT_INSERT, Constant Field Values

FORMATTER_INSERT_SPACE_AFTER_COMMA_IN_EXPLICIT_CONSTRUCTOR_CALL_ARGUMENTS

```
public static final String  FORMATTER_INSERT_SPACE_AFTER_COMMA_IN_EXPLICIT_CONSTRUCTOR_CALL_ARGUMENTS
```

```
 FORMATTER / Option to insert a space after the comma in the arguments of an explicit constructor call
     - option id:         "org.eclipse.jdt.core.formatter.insert_space_after_comma_in_explicitconstructorcall_arguments"
     - possible values:   { INSERT, DO_NOT_INSERT }
     - default:           INSERT
```

Since:

3.0

See Also:

JavaCore.INSERT, JavaCore.DO_NOT_INSERT, Constant Field Values

FORMATTER_INSERT_SPACE_AFTER_COMMA_IN_FOR_INCREMENTS

```
public static final String  FORMATTER_INSERT_SPACE_AFTER_COMMA_IN_FOR_INCREMENTS
```

```
 FORMATTER / Option to insert a space after the comma in the increments of a for statement
     - option id:         "org.eclipse.jdt.core.formatter.insert_space_after_comma_in_for_increments"
     - possible values:   { INSERT, DO_NOT_INSERT }
     - default:           INSERT
```

Since:

3.0

See Also:

JavaCore.INSERT, JavaCore.DO_NOT_INSERT, Constant Field Values

FORMATTER_INSERT_SPACE_AFTER_COMMA_IN_FOR_INITS

```
public static final String  FORMATTER_INSERT_SPACE_AFTER_COMMA_IN_FOR_INITS
```

```
 FORMATTER / Option to insert a space after the comma in the initializations of a for statement
     - option id:         "org.eclipse.jdt.core.formatter.insert_space_after_comma_in_for_inits"
     - possible values:   { INSERT, DO_NOT_INSERT }
     - default:           INSERT
```

Since:

3.0

See Also:

JavaCore.INSERT, JavaCore.DO_NOT_INSERT, Constant Field Values

FORMATTER_INSERT_SPACE_AFTER_COMMA_IN_METHOD_DECLARATION_PARAMETERS

```
public static final String  FORMATTER_INSERT_SPACE_AFTER_COMMA_IN_METHOD_DECLARATION_PARAMETERS
```

```
 FORMATTER / Option to insert a space after the comma in the parameters of a method declaration
     - option id:         "org.eclipse.jdt.core.formatter.insert_space_after_comma_in_method_declaration_parameters"
     - possible values:   { INSERT, DO_NOT_INSERT }
     - default:           INSERT
```

Since:

3.0

See Also:

JavaCore.INSERT, JavaCore.DO_NOT_INSERT, Constant Field Values

FORMATTER_INSERT_SPACE_AFTER_COMMA_IN_METHOD_DECLARATION_THROWS

```
public static final String  FORMATTER_INSERT_SPACE_AFTER_COMMA_IN_METHOD_DECLARATION_THROWS
```

```
 FORMATTER / Option to insert a space after the comma in the exception names in a throws clause of a method declaration
     - option id:         "org.eclipse.jdt.core.formatter.insert_space_after_comma_in_method_declaration_throws"
     - possible values:   { INSERT, DO_NOT_INSERT }
     - default:           INSERT
```

Since:

3.0

See Also:

JavaCore.INSERT, JavaCore.DO_NOT_INSERT, Constant Field Values

FORMATTER_INSERT_SPACE_AFTER_COMMA_IN_METHOD_INVOCATION_ARGUMENTS

```
public static final String  FORMATTER_INSERT_SPACE_AFTER_COMMA_IN_METHOD_INVOCATION_ARGUMENTS
```

```
 FORMATTER / Option to insert a space after the comma in the arguments of a method invocation
     - option id:         "org.eclipse.jdt.core.formatter.insert_space_after_comma_in_method_invocation_arguments"
     - possible values:   { INSERT, DO_NOT_INSERT }
     - default:           INSERT
```

Since:

3.0

See Also:

JavaCore.INSERT, JavaCore.DO_NOT_INSERT, Constant Field Values

FORMATTER_INSERT_SPACE_AFTER_COMMA_IN_MULTIPLE_FIELD_DECLARATIONS

```
public static final String  FORMATTER_INSERT_SPACE_AFTER_COMMA_IN_MULTIPLE_FIELD_DECLARATIONS
```

```
 FORMATTER / Option to insert a space after the comma in multiple field declaration
     - option id:         "org.eclipse.jdt.core.formatter.insert_space_after_comma_in_multiple_field_declarations"
     - possible values:   { INSERT, DO_NOT_INSERT }
     - default:           INSERT
```

Since:

3.0

See Also:

JavaCore.INSERT, JavaCore.DO_NOT_INSERT, Constant Field Values

FORMATTER_INSERT_SPACE_AFTER_COMMA_IN_MULTIPLE_LOCAL_DECLARATIONS

```
public static final String  FORMATTER_INSERT_SPACE_AFTER_COMMA_IN_MULTIPLE_LOCAL_DECLARATIONS
```

```
 FORMATTER / Option to insert a space after the comma in multiple local declaration
     - option id:         "org.eclipse.jdt.core.formatter.insert_space_after_comma_in_multiple_local_declarations"
     - possible values:   { INSERT, DO_NOT_INSERT }
     - default:           INSERT
```

Since:

3.0

See Also:

JavaCore.INSERT, JavaCore.DO_NOT_INSERT, Constant Field Values

FORMATTER_INSERT_SPACE_AFTER_COMMA_IN_PARAMETERIZED_TYPE_REFERENCE

```
public static final String  FORMATTER_INSERT_SPACE_AFTER_COMMA_IN_PARAMETERIZED_TYPE_REFERENCE
```

```
 FORMATTER / Option to insert a space after the comma in parameterized type reference
     - option id:         "org.eclipse.jdt.core.formatter.insert_space_after_comma_in_parameterized_type_reference"
     - possible values:   { INSERT, DO_NOT_INSERT }
     - default:           INSERT
```

Since:

3.1

See Also:

JavaCore.INSERT, JavaCore.DO_NOT_INSERT, Constant Field Values

FORMATTER_INSERT_SPACE_AFTER_COMMA_IN_PERMITTED_TYPES

```
public static final String  FORMATTER_INSERT_SPACE_AFTER_COMMA_IN_PERMITTED_TYPES
```

```
 FORMATTER / Option to insert a space after comma in the permitted types list in a type header
     - option id:         "org.eclipse.jdt.core.formatter.insert_space_after_comma_in_permitted_types"
     - possible values:   { INSERT, DO_NOT_INSERT }
     - default:           INSERT
```

Since:

3.29

See Also:

JavaCore.INSERT, JavaCore.DO_NOT_INSERT, Constant Field Values

FORMATTER_INSERT_SPACE_AFTER_COMMA_IN_RECORD_COMPONENTS

```
public static final String  FORMATTER_INSERT_SPACE_AFTER_COMMA_IN_RECORD_COMPONENTS
```

```
 FORMATTER / Option to insert a space after comma in record components list
     - option id:         "org.eclipse.jdt.core.formatter.insert_space_after_comma_in_record_components"
     - possible values:   { INSERT, DO_NOT_INSERT }
     - default:           INSERT
```

Since:

3.22

See Also:

JavaCore.INSERT, JavaCore.DO_NOT_INSERT, Constant Field Values

FORMATTER_INSERT_SPACE_AFTER_COMMA_IN_SUPERINTERFACES

```
public static final String  FORMATTER_INSERT_SPACE_AFTER_COMMA_IN_SUPERINTERFACES
```

```
 FORMATTER / Option to insert a space after the comma in superinterfaces names of a type header
     - option id:         "org.eclipse.jdt.core.formatter.insert_space_after_comma_in_superinterfaces"
     - possible values:   { INSERT, DO_NOT_INSERT }
     - default:           INSERT
```

Since:

3.0

See Also:

JavaCore.INSERT, JavaCore.DO_NOT_INSERT, Constant Field Values

FORMATTER_INSERT_SPACE_AFTER_COMMA_IN_SWITCH_CASE_EXPRESSIONS

```
public static final String  FORMATTER_INSERT_SPACE_AFTER_COMMA_IN_SWITCH_CASE_EXPRESSIONS
```

```
 FORMATTER / Option to insert a space after the comma in switch case expressions list
     - option id:         "org.eclipse.jdt.core.formatter.insert_space_after_comma_in_switch_case_expressions"
     - possible values:   { INSERT, DO_NOT_INSERT }
     - default:           INSERT
```

Since:

3.18

See Also:

JavaCore.INSERT, JavaCore.DO_NOT_INSERT, Constant Field Values

FORMATTER_INSERT_SPACE_AFTER_COMMA_IN_TYPE_ARGUMENTS

```
public static final String  FORMATTER_INSERT_SPACE_AFTER_COMMA_IN_TYPE_ARGUMENTS
```

```
 FORMATTER / Option to insert a space after the comma in explicit type arguments on method/constructor invocations
     - option id:         "org.eclipse.jdt.core.formatter.insert_space_after_comma_in_type_arguments"
     - possible values:   { INSERT, DO_NOT_INSERT }
     - default:           INSERT
```

Since:

3.1

See Also:

JavaCore.INSERT, JavaCore.DO_NOT_INSERT, Constant Field Values

FORMATTER_INSERT_SPACE_AFTER_COMMA_IN_TYPE_PARAMETERS

```
public static final String  FORMATTER_INSERT_SPACE_AFTER_COMMA_IN_TYPE_PARAMETERS
```

```
 FORMATTER / Option to insert a space after the comma in type parameter declarations
     - option id:         "org.eclipse.jdt.core.formatter.insert_space_after_comma_in_type_parameters"
     - possible values:   { INSERT, DO_NOT_INSERT }
     - default:           INSERT
```

Since:

3.1

See Also:

JavaCore.INSERT, JavaCore.DO_NOT_INSERT, Constant Field Values

FORMATTER_INSERT_SPACE_AFTER_ELLIPSIS

```
public static final String  FORMATTER_INSERT_SPACE_AFTER_ELLIPSIS
```

```
 FORMATTER / Option to insert a space after ellipsis
     - option id:         "org.eclipse.jdt.core.formatter.insert_space_after_ellipsis"
     - possible values:   { INSERT, DO_NOT_INSERT }
     - default:           INSERT
```

Since:

3.1

See Also:

JavaCore.INSERT, JavaCore.DO_NOT_INSERT, Constant Field Values

FORMATTER_INSERT_SPACE_AFTER_LAMBDA_ARROW

```
public static final String  FORMATTER_INSERT_SPACE_AFTER_LAMBDA_ARROW
```

```
 FORMATTER / Option to insert a space after the -> in lambda expressions
     - option id:         "org.eclipse.jdt.core.formatter.insert_space_after_lambda_arrow"
     - possible values:   { INSERT, DO_NOT_INSERT }
     - default:           INSERT
```

Since:

3.10

See Also:

JavaCore.INSERT, JavaCore.DO_NOT_INSERT, Constant Field Values

FORMATTER_INSERT_SPACE_AFTER_OPENING_ANGLE_BRACKET_IN_PARAMETERIZED_TYPE_REFERENCE

```
public static final String  FORMATTER_INSERT_SPACE_AFTER_OPENING_ANGLE_BRACKET_IN_PARAMETERIZED_TYPE_REFERENCE
```

```
 FORMATTER / Option to insert a space after the opening angle bracket in parameterized type reference
     - option id:         "org.eclipse.jdt.core.formatter.insert_space_after_opening_angle_bracket_in_parameterized_type_reference"
     - possible values:   { INSERT, DO_NOT_INSERT }
     - default:           DO_NOT_INSERT
```

Since:

3.1

See Also:

JavaCore.INSERT, JavaCore.DO_NOT_INSERT, Constant Field Values

FORMATTER_INSERT_SPACE_AFTER_OPENING_ANGLE_BRACKET_IN_TYPE_ARGUMENTS

```
public static final String  FORMATTER_INSERT_SPACE_AFTER_OPENING_ANGLE_BRACKET_IN_TYPE_ARGUMENTS
```

```
 FORMATTER / Option to insert a space after the opening angle bracket in explicit type arguments on method/constructor invocations
     - option id:         "org.eclipse.jdt.core.formatter.insert_space_after_opening_angle_bracket_in_type_arguments"
     - possible values:   { INSERT, DO_NOT_INSERT }
     - default:           DO_NOT_INSERT
```

Since:

3.1

See Also:

JavaCore.INSERT, JavaCore.DO_NOT_INSERT, Constant Field Values

FORMATTER_INSERT_SPACE_AFTER_OPENING_ANGLE_BRACKET_IN_TYPE_PARAMETERS

```
public static final String  FORMATTER_INSERT_SPACE_AFTER_OPENING_ANGLE_BRACKET_IN_TYPE_PARAMETERS
```

```
 FORMATTER / Option to insert a space after the opening angle bracket in type parameter declarations
     - option id:         "org.eclipse.jdt.core.formatter.insert_space_after_opening_angle_bracket_in_type_parameters"
     - possible values:   { INSERT, DO_NOT_INSERT }
     - default:           DO_NOT_INSERT
```

Since:

3.1

See Also:

JavaCore.INSERT, JavaCore.DO_NOT_INSERT, Constant Field Values

FORMATTER_INSERT_SPACE_AFTER_OPENING_BRACE_IN_ARRAY_INITIALIZER

```
public static final String  FORMATTER_INSERT_SPACE_AFTER_OPENING_BRACE_IN_ARRAY_INITIALIZER
```

```
 FORMATTER / Option to insert a space after the opening brace in an array initializer
     - option id:         "org.eclipse.jdt.core.formatter.insert_space_after_opening_brace_in_array_initializer"
     - possible values:   { INSERT, DO_NOT_INSERT }
     - default:           DO_NOT_INSERT
```

Since:

3.0

See Also:

JavaCore.INSERT, JavaCore.DO_NOT_INSERT, Constant Field Values

FORMATTER_INSERT_SPACE_AFTER_OPENING_BRACKET_IN_ARRAY_ALLOCATION_EXPRESSION

```
public static final String  FORMATTER_INSERT_SPACE_AFTER_OPENING_BRACKET_IN_ARRAY_ALLOCATION_EXPRESSION
```

```
 FORMATTER / Option to insert a space after the opening bracket inside an array allocation expression
     - option id:         "org.eclipse.jdt.core.formatter.insert_space_after_opening_bracket_in_array_allocation_expression"
     - possible values:   { INSERT, DO_NOT_INSERT }
     - default:           DO_NOT_INSERT
```

Since:

3.0

See Also:

JavaCore.INSERT, JavaCore.DO_NOT_INSERT, Constant Field Values

FORMATTER_INSERT_SPACE_AFTER_OPENING_BRACKET_IN_ARRAY_REFERENCE

```
public static final String  FORMATTER_INSERT_SPACE_AFTER_OPENING_BRACKET_IN_ARRAY_REFERENCE
```

```
 FORMATTER / Option to insert a space after the opening bracket inside an array reference
     - option id:         "org.eclipse.jdt.core.formatter.insert_space_after_opening_bracket_in_array_reference"
     - possible values:   { INSERT, DO_NOT_INSERT }
     - default:           DO_NOT_INSERT
```

Since:

3.0

See Also:

JavaCore.INSERT, JavaCore.DO_NOT_INSERT, Constant Field Values

FORMATTER_INSERT_SPACE_AFTER_OPENING_PAREN_IN_ANNOTATION

```
public static final String  FORMATTER_INSERT_SPACE_AFTER_OPENING_PAREN_IN_ANNOTATION
```

```
 FORMATTER / Option to insert a space after the opening parenthesis in annotation
     - option id:         "org.eclipse.jdt.core.formatter.insert_space_after_opening_paren_in_annotation"
     - possible values:   { INSERT, DO_NOT_INSERT }
     - default:           DO_NOT_INSERT
```

Since:

3.1

See Also:

JavaCore.INSERT, JavaCore.DO_NOT_INSERT, Constant Field Values

FORMATTER_INSERT_SPACE_AFTER_OPENING_PAREN_IN_CAST

```
public static final String  FORMATTER_INSERT_SPACE_AFTER_OPENING_PAREN_IN_CAST
```

```
 FORMATTER / Option to insert a space after the opening parenthesis in a cast expression
     - option id:         "org.eclipse.jdt.core.formatter.insert_space_after_opening_paren_in_cast"
     - possible values:   { INSERT, DO_NOT_INSERT }
     - default:           DO_NOT_INSERT
```

Since:

3.0

See Also:

JavaCore.INSERT, JavaCore.DO_NOT_INSERT, Constant Field Values

FORMATTER_INSERT_SPACE_AFTER_OPENING_PAREN_IN_CATCH

```
public static final String  FORMATTER_INSERT_SPACE_AFTER_OPENING_PAREN_IN_CATCH
```

```
 FORMATTER / Option to insert a space after the opening parenthesis in a catch
     - option id:         "org.eclipse.jdt.core.formatter.insert_space_after_opening_paren_in_catch"
     - possible values:   { INSERT, DO_NOT_INSERT }
     - default:           DO_NOT_INSERT
```

Since:

3.0

See Also:

JavaCore.INSERT, JavaCore.DO_NOT_INSERT, Constant Field Values

FORMATTER_INSERT_SPACE_AFTER_OPENING_PAREN_IN_CONSTRUCTOR_DECLARATION

```
public static final String  FORMATTER_INSERT_SPACE_AFTER_OPENING_PAREN_IN_CONSTRUCTOR_DECLARATION
```

```
 FORMATTER / Option to insert a space after the opening parenthesis in a constructor declaration
     - option id:         "org.eclipse.jdt.core.formatter.insert_space_after_opening_paren_in_constructor_declaration"
     - possible values:   { INSERT, DO_NOT_INSERT }
     - default:           DO_NOT_INSERT
```

Since:

3.0

See Also:

JavaCore.INSERT, JavaCore.DO_NOT_INSERT, Constant Field Values

FORMATTER_INSERT_SPACE_AFTER_OPENING_PAREN_IN_ENUM_CONSTANT

```
public static final String  FORMATTER_INSERT_SPACE_AFTER_OPENING_PAREN_IN_ENUM_CONSTANT
```

```
 FORMATTER / Option to insert a space after the opening parenthesis in enum constant
     - option id:         "org.eclipse.jdt.core.formatter.insert_space_after_opening_paren_in_enum_constant"
     - possible values:   { INSERT, DO_NOT_INSERT }
     - default:           DO_NOT_INSERT
```

Since:

3.1

See Also:

JavaCore.INSERT, JavaCore.DO_NOT_INSERT, Constant Field Values

FORMATTER_INSERT_SPACE_AFTER_OPENING_PAREN_IN_FOR

```
public static final String  FORMATTER_INSERT_SPACE_AFTER_OPENING_PAREN_IN_FOR
```

```
 FORMATTER / Option to insert a space after the opening parenthesis in a for statement
     - option id:         "org.eclipse.jdt.core.formatter.insert_space_after_opening_paren_in_for"
     - possible values:   { INSERT, DO_NOT_INSERT }
     - default:           DO_NOT_INSERT
```

Since:

3.0

See Also:

JavaCore.INSERT, JavaCore.DO_NOT_INSERT, Constant Field Values

FORMATTER_INSERT_SPACE_AFTER_OPENING_PAREN_IN_IF

```
public static final String  FORMATTER_INSERT_SPACE_AFTER_OPENING_PAREN_IN_IF
```

```
 FORMATTER / Option to insert a space after the opening parenthesis in an if statement
     - option id:         "org.eclipse.jdt.core.formatter.insert_space_after_opening_paren_in_if"
     - possible values:   { INSERT, DO_NOT_INSERT }
     - default:           DO_NOT_INSERT
```

Since:

3.0

See Also:

JavaCore.INSERT, JavaCore.DO_NOT_INSERT, Constant Field Values

FORMATTER_INSERT_SPACE_AFTER_OPENING_PAREN_IN_METHOD_DECLARATION

```
public static final String  FORMATTER_INSERT_SPACE_AFTER_OPENING_PAREN_IN_METHOD_DECLARATION
```

```
 FORMATTER / Option to insert a space after the opening parenthesis in a method declaration
     - option id:         "org.eclipse.jdt.core.formatter.insert_space_after_opening_paren_in_method_declaration"
     - possible values:   { INSERT, DO_NOT_INSERT }
     - default:           DO_NOT_INSERT
```

Since:

3.0

See Also:

JavaCore.INSERT, JavaCore.DO_NOT_INSERT, Constant Field Values

FORMATTER_INSERT_SPACE_AFTER_OPENING_PAREN_IN_METHOD_INVOCATION

```
public static final String  FORMATTER_INSERT_SPACE_AFTER_OPENING_PAREN_IN_METHOD_INVOCATION
```

```
 FORMATTER / Option to insert a space after the opening parenthesis in a method invocation
     - option id:         "org.eclipse.jdt.core.formatter.insert_space_after_opening_paren_in_method_invocation"
     - possible values:   { INSERT, DO_NOT_INSERT }
     - default:           DO_NOT_INSERT
```

Since:

3.0

See Also:

JavaCore.INSERT, JavaCore.DO_NOT_INSERT, Constant Field Values

FORMATTER_INSERT_SPACE_AFTER_OPENING_PAREN_IN_PARENTHESIZED_EXPRESSION

```
public static final String  FORMATTER_INSERT_SPACE_AFTER_OPENING_PAREN_IN_PARENTHESIZED_EXPRESSION
```

```
 FORMATTER / Option to insert a space after the opening parenthesis in a parenthesized expression
     - option id:         "org.eclipse.jdt.core.formatter.insert_space_after_opening_paren_in_parenthesized_expression"
     - possible values:   { INSERT, DO_NOT_INSERT }
     - default:           DO_NOT_INSERT
```

Since:

3.0

See Also:

JavaCore.INSERT, JavaCore.DO_NOT_INSERT, Constant Field Values

FORMATTER_INSERT_SPACE_AFTER_OPENING_PAREN_IN_RECORD_DECLARATION

```
public static final String  FORMATTER_INSERT_SPACE_AFTER_OPENING_PAREN_IN_RECORD_DECLARATION
```

```
 FORMATTER / Option to insert a space after the opening parenthesis in a record declaration
     - option id:         "org.eclipse.jdt.core.formatter.insert_space_after_opening_paren_in_record_declaration"
     - possible values:   { INSERT, DO_NOT_INSERT }
     - default:           DO_NOT_INSERT
```

Since:

3.22

See Also:

JavaCore.INSERT, JavaCore.DO_NOT_INSERT, Constant Field Values

FORMATTER_INSERT_SPACE_AFTER_OPENING_PAREN_IN_SWITCH

```
public static final String  FORMATTER_INSERT_SPACE_AFTER_OPENING_PAREN_IN_SWITCH
```

```
 FORMATTER / Option to insert a space after the opening parenthesis in a switch statement
     - option id:         "org.eclipse.jdt.core.formatter.insert_space_after_opening_paren_in_switch"
     - possible values:   { INSERT, DO_NOT_INSERT }
     - default:           DO_NOT_INSERT
```

Since:

3.0

See Also:

JavaCore.INSERT, JavaCore.DO_NOT_INSERT, Constant Field Values

FORMATTER_INSERT_SPACE_AFTER_OPENING_PAREN_IN_SYNCHRONIZED

```
public static final String  FORMATTER_INSERT_SPACE_AFTER_OPENING_PAREN_IN_SYNCHRONIZED
```

```
 FORMATTER / Option to insert a space after the opening parenthesis in a synchronized statement
     - option id:         "org.eclipse.jdt.core.formatter.insert_space_after_opening_paren_in_synchronized"
     - possible values:   { INSERT, DO_NOT_INSERT }
     - default:           DO_NOT_INSERT
```

Since:

3.0

See Also:

JavaCore.INSERT, JavaCore.DO_NOT_INSERT, Constant Field Values

FORMATTER_INSERT_SPACE_AFTER_OPENING_PAREN_IN_TRY

```
public static final String  FORMATTER_INSERT_SPACE_AFTER_OPENING_PAREN_IN_TRY
```

```
 FORMATTER / Option to insert a space after the opening parenthesis in a try with resources statement
     - option id:         "org.eclipse.jdt.core.formatter.insert_space_after_opening_paren_in_try"
     - possible values:   { INSERT, DO_NOT_INSERT }
     - default:           DO_NOT_INSERT
```

Since:

3.7.1

See Also:

JavaCore.INSERT, JavaCore.DO_NOT_INSERT, Constant Field Values

FORMATTER_INSERT_SPACE_AFTER_OPENING_PAREN_IN_WHILE

```
public static final String  FORMATTER_INSERT_SPACE_AFTER_OPENING_PAREN_IN_WHILE
```

```
 FORMATTER / Option to insert a space after the opening parenthesis in a while statement
     - option id:         "org.eclipse.jdt.core.formatter.insert_space_after_opening_paren_in_while"
     - possible values:   { INSERT, DO_NOT_INSERT }
     - default:           DO_NOT_INSERT
```

Since:

3.0

See Also:

JavaCore.INSERT, JavaCore.DO_NOT_INSERT, Constant Field Values

FORMATTER_INSERT_SPACE_AFTER_POSTFIX_OPERATOR

```
public static final String  FORMATTER_INSERT_SPACE_AFTER_POSTFIX_OPERATOR
```

```
 FORMATTER / Option to insert a space after a postfix operator
     - option id:         "org.eclipse.jdt.core.formatter.insert_space_after_postfix_operator"
     - possible values:   { INSERT, DO_NOT_INSERT }
     - default:           DO_NOT_INSERT
```

Since:

3.0

See Also:

JavaCore.INSERT, JavaCore.DO_NOT_INSERT, Constant Field Values

FORMATTER_INSERT_SPACE_AFTER_PREFIX_OPERATOR

```
public static final String  FORMATTER_INSERT_SPACE_AFTER_PREFIX_OPERATOR
```

```
 FORMATTER / Option to insert a space after a prefix operator
     - option id:         "org.eclipse.jdt.core.formatter.insert_space_after_prefix_operator"
     - possible values:   { INSERT, DO_NOT_INSERT }
     - default:           DO_NOT_INSERT
```

Since:

3.0

See Also:

JavaCore.INSERT, JavaCore.DO_NOT_INSERT, Constant Field Values

FORMATTER_INSERT_SPACE_AFTER_QUESTION_IN_CONDITIONAL

```
public static final String  FORMATTER_INSERT_SPACE_AFTER_QUESTION_IN_CONDITIONAL
```

```
 FORMATTER / Option to insert a space after question mark in a conditional expression
     - option id:         "org.eclipse.jdt.core.formatter.insert_space_after_question_in_conditional"
     - possible values:   { INSERT, DO_NOT_INSERT }
     - default:           DO_NOT_INSERT
```

Since:

3.0

See Also:

JavaCore.INSERT, JavaCore.DO_NOT_INSERT, Constant Field Values

FORMATTER_INSERT_SPACE_AFTER_QUESTION_IN_WILDCARD

```
public static final String  FORMATTER_INSERT_SPACE_AFTER_QUESTION_IN_WILDCARD
```

```
 FORMATTER / Option to insert a space after question mark in a wildcard
     - option id:         "org.eclipse.jdt.core.formatter.insert_space_after_question_in_wildcard"
     - possible values:   { INSERT, DO_NOT_INSERT }
     - default:           DO_NOT_INSERT
```

Since:

3.1

See Also:

JavaCore.INSERT, JavaCore.DO_NOT_INSERT, Constant Field Values

FORMATTER_INSERT_SPACE_AFTER_SEMICOLON_IN_FOR

```
public static final String  FORMATTER_INSERT_SPACE_AFTER_SEMICOLON_IN_FOR
```

```
 FORMATTER / Option to insert a space after semicolon in a for statement
     - option id:         "org.eclipse.jdt.core.formatter.insert_space_after_semicolon_in_for"
     - possible values:   { INSERT, DO_NOT_INSERT }
     - default:           INSERT
```

Since:

3.0

See Also:

JavaCore.INSERT, JavaCore.DO_NOT_INSERT, Constant Field Values

FORMATTER_INSERT_SPACE_AFTER_SEMICOLON_IN_TRY_RESOURCES

```
public static final String  FORMATTER_INSERT_SPACE_AFTER_SEMICOLON_IN_TRY_RESOURCES
```

```
 FORMATTER / Option to insert a space after semicolons following each resource declaration in a try with
 resources statement
     - option id:         "org.eclipse.jdt.core.formatter.insert_space_after_semicolon_in_try_resources"
     - possible values:   { INSERT, DO_NOT_INSERT }
     - default:           INSERT
```

Since:

3.7.1

See Also:

JavaCore.INSERT, JavaCore.DO_NOT_INSERT, Constant Field Values

FORMATTER_INSERT_SPACE_AFTER_UNARY_OPERATOR

```
public static final String  FORMATTER_INSERT_SPACE_AFTER_UNARY_OPERATOR
```

```
 FORMATTER / Option to insert a space after an unary operator
     - option id:         "org.eclipse.jdt.core.formatter.insert_space_after_unary_operator"
     - possible values:   { INSERT, DO_NOT_INSERT }
     - default:           DO_NOT_INSERT
```

Since:

3.0

See Also:

JavaCore.INSERT, JavaCore.DO_NOT_INSERT, Constant Field Values

FORMATTER_INSERT_SPACE_AFTER_NOT_OPERATOR

```
public static final String  FORMATTER_INSERT_SPACE_AFTER_NOT_OPERATOR
```

```
 FORMATTER / Option to insert a space after 'not' operator
     - option id:         "org.eclipse.jdt.core.formatter.insert_space_after_not_operator"
     - possible values:   { INSERT, DO_NOT_INSERT }
     - default:           DO_NOT_INSERT
```

Since:

3.20

See Also:

JavaCore.INSERT, JavaCore.DO_NOT_INSERT, Constant Field Values

FORMATTER_INSERT_SPACE_BEFORE_AND_IN_TYPE_PARAMETER

```
public static final String  FORMATTER_INSERT_SPACE_BEFORE_AND_IN_TYPE_PARAMETER
```

```
 FORMATTER / Option to insert a space before and in wildcard
     - option id:         "org.eclipse.jdt.core.formatter.insert_space_before_and_in_type_parameter"
     - possible values:   { INSERT, DO_NOT_INSERT }
     - default:           INSERT
```

Since:

3.1

See Also:

JavaCore.INSERT, JavaCore.DO_NOT_INSERT, Constant Field Values

FORMATTER_INSERT_SPACE_BEFORE_ARROW_IN_SWITCH_CASE

```
public static final String  FORMATTER_INSERT_SPACE_BEFORE_ARROW_IN_SWITCH_CASE
```

```
 FORMATTER / Option to insert a space before arrow in switch case
     - option id:         "org.eclipse.jdt.core.formatter.insert_space_before_arrow_in_switch_case"
     - possible values:   { INSERT, DO_NOT_INSERT }
     - default:           INSERT
```

Since:

3.18

See Also:

JavaCore.INSERT, JavaCore.DO_NOT_INSERT, Constant Field Values

FORMATTER_INSERT_SPACE_BEFORE_ARROW_IN_SWITCH_DEFAULT

```
public static final String  FORMATTER_INSERT_SPACE_BEFORE_ARROW_IN_SWITCH_DEFAULT
```

```
 FORMATTER / Option to insert a space before arrow in switch default
     - option id:         "org.eclipse.jdt.core.formatter.insert_space_before_arrow_in_switch_default"
     - possible values:   { INSERT, DO_NOT_INSERT }
     - default:           INSERT
```

Since:

3.18

See Also:

JavaCore.INSERT, JavaCore.DO_NOT_INSERT, Constant Field Values

FORMATTER_INSERT_SPACE_BEFORE_ASSIGNMENT_OPERATOR

```
public static final String  FORMATTER_INSERT_SPACE_BEFORE_ASSIGNMENT_OPERATOR
```

```
 FORMATTER / Option to insert a space before an assignment operator
     - option id:         "org.eclipse.jdt.core.formatter.insert_space_before_assignment_operator"
     - possible values:   { INSERT, DO_NOT_INSERT }
     - default:           INSERT
```

Since:

3.0

See Also:

JavaCore.INSERT, JavaCore.DO_NOT_INSERT, Constant Field Values

FORMATTER_INSERT_SPACE_BEFORE_AT_IN_ANNOTATION_TYPE_DECLARATION

```
public static final String  FORMATTER_INSERT_SPACE_BEFORE_AT_IN_ANNOTATION_TYPE_DECLARATION
```

```
 FORMATTER / Option to insert a space before at in annotation type declaration
     - option id:         "org.eclipse.jdt.core.formatter.insert_space_before_at_in_annotation_type_declaration"
     - possible values:   { INSERT, DO_NOT_INSERT }
     - default:           INSERT
```

Since:

3.1

See Also:

JavaCore.INSERT, JavaCore.DO_NOT_INSERT, Constant Field Values

FORMATTER_INSERT_SPACE_BEFORE_BINARY_OPERATOR

```
public static final String  FORMATTER_INSERT_SPACE_BEFORE_BINARY_OPERATOR
```

Deprecated.
Use the new settings instead: FORMATTER_INSERT_SPACE_BEFORE_MULTIPLICATIVE_OPERATOR, FORMATTER_INSERT_SPACE_BEFORE_ADDITIVE_OPERATOR, FORMATTER_INSERT_SPACE_BEFORE_STRING_CONCATENATION, FORMATTER_INSERT_SPACE_BEFORE_SHIFT_OPERATOR, FORMATTER_INSERT_SPACE_BEFORE_RELATIONAL_OPERATOR, FORMATTER_INSERT_SPACE_BEFORE_BITWISE_OPERATOR, FORMATTER_INSERT_SPACE_BEFORE_LOGICAL_OPERATOR

```
 FORMATTER / Option to insert a space before an binary operator
     - option id:         "org.eclipse.jdt.core.formatter.insert_space_before_binary_operator"
     - possible values:   { INSERT, DO_NOT_INSERT }
     - default:           DO_NOT_INSERT
```

Since:

3.0

See Also:

JavaCore.INSERT, JavaCore.DO_NOT_INSERT, Constant Field Values

FORMATTER_INSERT_SPACE_BEFORE_MULTIPLICATIVE_OPERATOR

```
public static final String  FORMATTER_INSERT_SPACE_BEFORE_MULTIPLICATIVE_OPERATOR
```

```
 FORMATTER / Option to insert a space before a multiplicative operator (*, /, %)
     - option id:         "org.eclipse.jdt.core.formatter.insert_space_before_multiplicative_operator"
     - possible values:   { INSERT, DO_NOT_INSERT }
     - default:           INSERT
```

Since:

3.17

See Also:

JavaCore.INSERT, JavaCore.DO_NOT_INSERT, Constant Field Values

FORMATTER_INSERT_SPACE_BEFORE_ADDITIVE_OPERATOR

```
public static final String  FORMATTER_INSERT_SPACE_BEFORE_ADDITIVE_OPERATOR
```

```
 FORMATTER / Option to insert a space before an additive operator (+, -)
     - option id:         "org.eclipse.jdt.core.formatter.insert_space_before_additive_operator"
     - possible values:   { INSERT, DO_NOT_INSERT }
     - default:           INSERT
```

Since:

3.17

See Also:

JavaCore.INSERT, JavaCore.DO_NOT_INSERT, Constant Field Values

FORMATTER_INSERT_SPACE_BEFORE_STRING_CONCATENATION

```
public static final String  FORMATTER_INSERT_SPACE_BEFORE_STRING_CONCATENATION
```

```
 FORMATTER / Option to insert a space before a string concatenation operator
     - option id:         "org.eclipse.jdt.core.formatter.insert_space_before_string_concatenation"
     - possible values:   { INSERT, DO_NOT_INSERT }
     - default:           INSERT
```

Since:

3.17

See Also:

JavaCore.INSERT, JavaCore.DO_NOT_INSERT, Constant Field Values

FORMATTER_INSERT_SPACE_BEFORE_SHIFT_OPERATOR

```
public static final String  FORMATTER_INSERT_SPACE_BEFORE_SHIFT_OPERATOR
```

```
 FORMATTER / Option to insert a space before a shift operator (<<, >>, >>>)
     - option id:         "org.eclipse.jdt.core.formatter.insert_space_before_shift_operator"
     - possible values:   { INSERT, DO_NOT_INSERT }
     - default:           INSERT
```

Since:

3.17

See Also:

JavaCore.INSERT, JavaCore.DO_NOT_INSERT, Constant Field Values

FORMATTER_INSERT_SPACE_BEFORE_RELATIONAL_OPERATOR

```
public static final String  FORMATTER_INSERT_SPACE_BEFORE_RELATIONAL_OPERATOR
```

```
 FORMATTER / Option to insert a space before a relational operator (<, >, <=, >=, ==, !=)
     - option id:         "org.eclipse.jdt.core.formatter.insert_space_before_relational_operator"
     - possible values:   { INSERT, DO_NOT_INSERT }
     - default:           INSERT
```

Since:

3.17

See Also:

JavaCore.INSERT, JavaCore.DO_NOT_INSERT, Constant Field Values

FORMATTER_INSERT_SPACE_BEFORE_BITWISE_OPERATOR

```
public static final String  FORMATTER_INSERT_SPACE_BEFORE_BITWISE_OPERATOR
```

```
 FORMATTER / Option to insert a space before a bitwise operator (&, ^, |)
     - option id:         "org.eclipse.jdt.core.formatter.insert_space_before_bitwise_operator"
     - possible values:   { INSERT, DO_NOT_INSERT }
     - default:           INSERT
```

Since:

3.17

See Also:

JavaCore.INSERT, JavaCore.DO_NOT_INSERT, Constant Field Values

FORMATTER_INSERT_SPACE_BEFORE_LOGICAL_OPERATOR

```
public static final String  FORMATTER_INSERT_SPACE_BEFORE_LOGICAL_OPERATOR
```

```
 FORMATTER / Option to insert a space before a logical operator (&&, ||)
     - option id:         "org.eclipse.jdt.core.formatter.insert_space_before_logical_operator"
     - possible values:   { INSERT, DO_NOT_INSERT }
     - default:           INSERT
```

Since:

3.17

See Also:

JavaCore.INSERT, JavaCore.DO_NOT_INSERT, Constant Field Values

FORMATTER_INSERT_SPACE_BEFORE_CLOSING_ANGLE_BRACKET_IN_PARAMETERIZED_TYPE_REFERENCE

```
public static final String  FORMATTER_INSERT_SPACE_BEFORE_CLOSING_ANGLE_BRACKET_IN_PARAMETERIZED_TYPE_REFERENCE
```

```
 FORMATTER / Option to insert a space before the closing angle bracket in parameterized type reference
     - option id:         "org.eclipse.jdt.core.formatter.insert_space_before_closing_angle_bracket_in_parameterized_type_reference"
     - possible values:   { INSERT, DO_NOT_INSERT }
     - default:           DO_NOT_INSERT
```

Since:

3.1

See Also:

JavaCore.INSERT, JavaCore.DO_NOT_INSERT, Constant Field Values

FORMATTER_INSERT_SPACE_BEFORE_CLOSING_ANGLE_BRACKET_IN_TYPE_ARGUMENTS

```
public static final String  FORMATTER_INSERT_SPACE_BEFORE_CLOSING_ANGLE_BRACKET_IN_TYPE_ARGUMENTS
```

```
 FORMATTER / Option to insert a space before the closing angle bracket in explicit type arguments on method/constructor invocations
     - option id:         "org.eclipse.jdt.core.formatter.insert_space_before_closing_angle_bracket_in_type_arguments"
     - possible values:   { INSERT, DO_NOT_INSERT }
     - default:           DO_NOT_INSERT
```

Since:

3.1

See Also:

JavaCore.INSERT, JavaCore.DO_NOT_INSERT, Constant Field Values

FORMATTER_INSERT_SPACE_BEFORE_CLOSING_ANGLE_BRACKET_IN_TYPE_PARAMETERS

```
public static final String  FORMATTER_INSERT_SPACE_BEFORE_CLOSING_ANGLE_BRACKET_IN_TYPE_PARAMETERS
```

```
 FORMATTER / Option to insert a space before the closing angle bracket in type parameter declarations
     - option id:         "org.eclipse.jdt.core.formatter.insert_space_before_closing_angle_bracket_in_type_parameters"
     - possible values:   { INSERT, DO_NOT_INSERT }
     - default:           DO_NOT_INSERT
```

Since:

3.1

See Also:

JavaCore.INSERT, JavaCore.DO_NOT_INSERT, Constant Field Values

FORMATTER_INSERT_SPACE_BEFORE_CLOSING_BRACE_IN_ARRAY_INITIALIZER

```
public static final String  FORMATTER_INSERT_SPACE_BEFORE_CLOSING_BRACE_IN_ARRAY_INITIALIZER
```

```
 FORMATTER / Option to insert a space before the closing brace in an array initializer
     - option id:         "org.eclipse.jdt.core.formatter.insert_space_before_closing_brace_in_array_initializer"
     - possible values:   { INSERT, DO_NOT_INSERT }
     - default:           DO_NOT_INSERT
```

Since:

3.0

See Also:

JavaCore.INSERT, JavaCore.DO_NOT_INSERT, Constant Field Values

FORMATTER_INSERT_SPACE_BEFORE_CLOSING_BRACKET_IN_ARRAY_ALLOCATION_EXPRESSION

```
public static final String  FORMATTER_INSERT_SPACE_BEFORE_CLOSING_BRACKET_IN_ARRAY_ALLOCATION_EXPRESSION
```

```
 FORMATTER / Option to insert a space before the closing bracket in an array allocation expression
     - option id:         "org.eclipse.jdt.core.formatter.insert_space_before_closing_bracket_in_array_allocation_expression"
     - possible values:   { INSERT, DO_NOT_INSERT }
     - default:           DO_NOT_INSERT
```

Since:

3.0

See Also:

JavaCore.INSERT, JavaCore.DO_NOT_INSERT, Constant Field Values

FORMATTER_INSERT_SPACE_BEFORE_CLOSING_BRACKET_IN_ARRAY_REFERENCE

```
public static final String  FORMATTER_INSERT_SPACE_BEFORE_CLOSING_BRACKET_IN_ARRAY_REFERENCE
```

```
 FORMATTER / Option to insert a space before the closing bracket in an array reference
     - option id:         "org.eclipse.jdt.core.formatter.insert_space_before_closing_bracket_in_array_reference"
     - possible values:   { INSERT, DO_NOT_INSERT }
     - default:           DO_NOT_INSERT
```

Since:

3.0

See Also:

JavaCore.INSERT, JavaCore.DO_NOT_INSERT, Constant Field Values

FORMATTER_INSERT_SPACE_BEFORE_CLOSING_PAREN_IN_ANNOTATION

```
public static final String  FORMATTER_INSERT_SPACE_BEFORE_CLOSING_PAREN_IN_ANNOTATION
```

```
 FORMATTER / Option to insert a space before the closing parenthesis in annotation
     - option id:         "org.eclipse.jdt.core.formatter.insert_space_before_closing_paren_in_annotation"
     - possible values:   { INSERT, DO_NOT_INSERT }
     - default:           DO_NOT_INSERT
```

Since:

3.1

See Also:

JavaCore.INSERT, JavaCore.DO_NOT_INSERT, Constant Field Values

FORMATTER_INSERT_SPACE_BEFORE_CLOSING_PAREN_IN_CAST

```
public static final String  FORMATTER_INSERT_SPACE_BEFORE_CLOSING_PAREN_IN_CAST
```

```
 FORMATTER / Option to insert a space before the closing parenthesis in a cast expression
     - option id:         "org.eclipse.jdt.core.formatter.insert_space_before_closing_paren_in_cast"
     - possible values:   { INSERT, DO_NOT_INSERT }
     - default:           DO_NOT_INSERT
```

Since:

3.0

See Also:

JavaCore.INSERT, JavaCore.DO_NOT_INSERT, Constant Field Values

FORMATTER_INSERT_SPACE_BEFORE_CLOSING_PAREN_IN_CATCH

```
public static final String  FORMATTER_INSERT_SPACE_BEFORE_CLOSING_PAREN_IN_CATCH
```

```
 FORMATTER / Option to insert a space before the closing parenthesis in a catch
     - option id:         "org.eclipse.jdt.core.formatter.insert_space_before_closing_paren_in_catch"
     - possible values:   { INSERT, DO_NOT_INSERT }
     - default:           DO_NOT_INSERT
```

Since:

3.0

See Also:

JavaCore.INSERT, JavaCore.DO_NOT_INSERT, Constant Field Values

FORMATTER_INSERT_SPACE_BEFORE_CLOSING_PAREN_IN_CONSTRUCTOR_DECLARATION

```
public static final String  FORMATTER_INSERT_SPACE_BEFORE_CLOSING_PAREN_IN_CONSTRUCTOR_DECLARATION
```

```
 FORMATTER / Option to insert a space before the closing parenthesis in a constructor declaration
     - option id:         "org.eclipse.jdt.core.formatter.insert_space_before_closing_paren_in_constructor_declaration"
     - possible values:   { INSERT, DO_NOT_INSERT }
     - default:           DO_NOT_INSERT
```

Since:

3.0

See Also:

JavaCore.INSERT, JavaCore.DO_NOT_INSERT, Constant Field Values

FORMATTER_INSERT_SPACE_BEFORE_CLOSING_PAREN_IN_ENUM_CONSTANT

```
public static final String  FORMATTER_INSERT_SPACE_BEFORE_CLOSING_PAREN_IN_ENUM_CONSTANT
```

```
 FORMATTER / Option to insert a space before the closing parenthesis in enum constant
     - option id:         "org.eclipse.jdt.core.formatter.insert_space_before_closing_paren_in_enum_constant"
     - possible values:   { INSERT, DO_NOT_INSERT }
     - default:           DO_NOT_INSERT
```

Since:

3.1

See Also:

JavaCore.INSERT, JavaCore.DO_NOT_INSERT, Constant Field Values

FORMATTER_INSERT_SPACE_BEFORE_CLOSING_PAREN_IN_FOR

```
public static final String  FORMATTER_INSERT_SPACE_BEFORE_CLOSING_PAREN_IN_FOR
```

```
 FORMATTER / Option to insert a space before the closing parenthesis in a for statement
     - option id:         "org.eclipse.jdt.core.formatter.insert_space_before_closing_paren_in_for"
     - possible values:   { INSERT, DO_NOT_INSERT }
     - default:           DO_NOT_INSERT
```

Since:

3.0

See Also:

JavaCore.INSERT, JavaCore.DO_NOT_INSERT, Constant Field Values

FORMATTER_INSERT_SPACE_BEFORE_CLOSING_PAREN_IN_IF

```
public static final String  FORMATTER_INSERT_SPACE_BEFORE_CLOSING_PAREN_IN_IF
```

```
 FORMATTER / Option to insert a space before the closing parenthesis in an if statement
     - option id:         "org.eclipse.jdt.core.formatter.insert_space_before_closing_paren_in_if"
     - possible values:   { INSERT, DO_NOT_INSERT }
     - default:           DO_NOT_INSERT
```

Since:

3.0

See Also:

JavaCore.INSERT, JavaCore.DO_NOT_INSERT, Constant Field Values

FORMATTER_INSERT_SPACE_BEFORE_CLOSING_PAREN_IN_METHOD_DECLARATION

```
public static final String  FORMATTER_INSERT_SPACE_BEFORE_CLOSING_PAREN_IN_METHOD_DECLARATION
```

```
 FORMATTER / Option to insert a space before the closing parenthesis in a method declaration
     - option id:         "org.eclipse.jdt.core.formatter.insert_space_before_closing_paren_in_method_declaration"
     - possible values:   { INSERT, DO_NOT_INSERT }
     - default:           DO_NOT_INSERT
```

Since:

3.0

See Also:

JavaCore.INSERT, JavaCore.DO_NOT_INSERT, Constant Field Values

FORMATTER_INSERT_SPACE_BEFORE_CLOSING_PAREN_IN_METHOD_INVOCATION

```
public static final String  FORMATTER_INSERT_SPACE_BEFORE_CLOSING_PAREN_IN_METHOD_INVOCATION
```

```
 FORMATTER / Option to insert a space before the closing parenthesis in a method invocation
     - option id:         "org.eclipse.jdt.core.formatter.insert_space_before_closing_paren_in_method_invocation"
     - possible values:   { INSERT, DO_NOT_INSERT }
     - default:           DO_NOT_INSERT
```

Since:

3.0

See Also:

JavaCore.INSERT, JavaCore.DO_NOT_INSERT, Constant Field Values

FORMATTER_INSERT_SPACE_BEFORE_CLOSING_PAREN_IN_PARENTHESIZED_EXPRESSION

```
public static final String  FORMATTER_INSERT_SPACE_BEFORE_CLOSING_PAREN_IN_PARENTHESIZED_EXPRESSION
```

```
 FORMATTER / Option to insert a space before the closing parenthesis in a parenthesized expression
     - option id:         "org.eclipse.jdt.core.formatter.insert_space_before_closing_paren_in_parenthesized_expression"
     - possible values:   { INSERT, DO_NOT_INSERT }
     - default:           DO_NOT_INSERT
```

Since:

3.0

See Also:

JavaCore.INSERT, JavaCore.DO_NOT_INSERT, Constant Field Values

FORMATTER_INSERT_SPACE_BEFORE_CLOSING_PAREN_IN_RECORD_DECLARATION

```
public static final String  FORMATTER_INSERT_SPACE_BEFORE_CLOSING_PAREN_IN_RECORD_DECLARATION
```

```
 FORMATTER / Option to insert a space before the closing parenthesis in a record declaration
     - option id:         "org.eclipse.jdt.core.formatter.insert_space_before_closing_paren_in_record_declaration"
     - possible values:   { INSERT, DO_NOT_INSERT }
     - default:           DO_NOT_INSERT
```

Since:

3.22

See Also:

JavaCore.INSERT, JavaCore.DO_NOT_INSERT, Constant Field Values

FORMATTER_INSERT_SPACE_BEFORE_CLOSING_PAREN_IN_SWITCH

```
public static final String  FORMATTER_INSERT_SPACE_BEFORE_CLOSING_PAREN_IN_SWITCH
```

```
 FORMATTER / Option to insert a space before the closing parenthesis in a switch statement
     - option id:         "org.eclipse.jdt.core.formatter.insert_space_before_closing_paren_in_switch"
     - possible values:   { INSERT, DO_NOT_INSERT }
     - default:           DO_NOT_INSERT
```

Since:

3.0

See Also:

JavaCore.INSERT, JavaCore.DO_NOT_INSERT, Constant Field Values

FORMATTER_INSERT_SPACE_BEFORE_CLOSING_PAREN_IN_SYNCHRONIZED

```
public static final String  FORMATTER_INSERT_SPACE_BEFORE_CLOSING_PAREN_IN_SYNCHRONIZED
```

```
 FORMATTER / Option to insert a space before the closing parenthesis in a synchronized statement
     - option id:         "org.eclipse.jdt.core.formatter.insert_space_before_closing_paren_in_synchronized"
     - possible values:   { INSERT, DO_NOT_INSERT }
     - default:           DO_NOT_INSERT
```

Since:

3.0

See Also:

JavaCore.INSERT, JavaCore.DO_NOT_INSERT, Constant Field Values

FORMATTER_INSERT_SPACE_BEFORE_CLOSING_PAREN_IN_TRY

```
public static final String  FORMATTER_INSERT_SPACE_BEFORE_CLOSING_PAREN_IN_TRY
```

```
 FORMATTER / Option to insert a space before the closing parenthesis in a try with resources statement
     - option id:         "org.eclipse.jdt.core.formatter.insert_space_before_closing_paren_in_try"
     - possible values:   { INSERT, DO_NOT_INSERT }
     - default:           DO_NOT_INSERT
```

Since:

3.7.1

See Also:

JavaCore.INSERT, JavaCore.DO_NOT_INSERT, Constant Field Values

FORMATTER_INSERT_SPACE_BEFORE_CLOSING_PAREN_IN_WHILE

```
public static final String  FORMATTER_INSERT_SPACE_BEFORE_CLOSING_PAREN_IN_WHILE
```

```
 FORMATTER / Option to insert a space before the closing parenthesis in a while statement
     - option id:         "org.eclipse.jdt.core.formatter.insert_space_before_closing_paren_in_while"
     - possible values:   { INSERT, DO_NOT_INSERT }
     - default:           DO_NOT_INSERT
```

Since:

3.0

See Also:

JavaCore.INSERT, JavaCore.DO_NOT_INSERT, Constant Field Values

FORMATTER_INSERT_SPACE_BEFORE_COLON_IN_ASSERT

```
public static final String  FORMATTER_INSERT_SPACE_BEFORE_COLON_IN_ASSERT
```

```
 FORMATTER / Option to insert a space before colon in an assert statement
     - option id:         "org.eclipse.jdt.core.formatter.insert_space_before_colon_in_assert"
     - possible values:   { INSERT, DO_NOT_INSERT }
     - default:           INSERT
```

Since:

3.0

See Also:

JavaCore.INSERT, JavaCore.DO_NOT_INSERT, Constant Field Values

FORMATTER_INSERT_SPACE_BEFORE_COLON_IN_CASE

```
public static final String  FORMATTER_INSERT_SPACE_BEFORE_COLON_IN_CASE
```

```
 FORMATTER / Option to insert a space before colon in a case statement
     - option id:         "org.eclipse.jdt.core.formatter.insert_space_before_colon_in_case"
     - possible values:   { INSERT, DO_NOT_INSERT }
     - default:           INSERT
```

Since:

3.0

See Also:

JavaCore.INSERT, JavaCore.DO_NOT_INSERT, Constant Field Values

FORMATTER_INSERT_SPACE_BEFORE_COLON_IN_CONDITIONAL

```
public static final String  FORMATTER_INSERT_SPACE_BEFORE_COLON_IN_CONDITIONAL
```

```
 FORMATTER / Option to insert a space before colon in a conditional expression
     - option id:         "org.eclipse.jdt.core.formatter.insert_space_before_colon_in_conditional"
     - possible values:   { INSERT, DO_NOT_INSERT }
     - default:           INSERT
```

Since:

3.0

See Also:

JavaCore.INSERT, JavaCore.DO_NOT_INSERT, Constant Field Values

FORMATTER_INSERT_SPACE_BEFORE_COLON_IN_DEFAULT

```
public static final String  FORMATTER_INSERT_SPACE_BEFORE_COLON_IN_DEFAULT
```

```
 FORMATTER / Option to insert a space before colon in a default statement
     - option id:         "org.eclipse.jdt.core.formatter.insert_space_before_colon_in_default"
     - possible values:   { INSERT, DO_NOT_INSERT }
     - default:           INSERT
```

Since:

3.0

See Also:

JavaCore.INSERT, JavaCore.DO_NOT_INSERT, Constant Field Values

FORMATTER_INSERT_SPACE_BEFORE_COLON_IN_FOR

```
public static final String  FORMATTER_INSERT_SPACE_BEFORE_COLON_IN_FOR
```

```
 FORMATTER / Option to insert a space before colon in a for statement
     - option id:         "org.eclipse.jdt.core.formatter.insert_space_before_colon_in_for"
     - possible values:   { INSERT, DO_NOT_INSERT }
     - default:           INSERT
```

Since:

3.1

See Also:

JavaCore.INSERT, JavaCore.DO_NOT_INSERT, Constant Field Values

FORMATTER_INSERT_SPACE_BEFORE_COLON_IN_LABELED_STATEMENT

```
public static final String  FORMATTER_INSERT_SPACE_BEFORE_COLON_IN_LABELED_STATEMENT
```

```
 FORMATTER / Option to insert a space before colon in a labeled statement
     - option id:         "org.eclipse.jdt.core.formatter.insert_space_before_colon_in_labeled_statement"
     - possible values:   { INSERT, DO_NOT_INSERT }
     - default:           INSERT
```

Since:

3.0

See Also:

JavaCore.INSERT, JavaCore.DO_NOT_INSERT, Constant Field Values

FORMATTER_INSERT_SPACE_BEFORE_COMMA_IN_ALLOCATION_EXPRESSION

```
public static final String  FORMATTER_INSERT_SPACE_BEFORE_COMMA_IN_ALLOCATION_EXPRESSION
```

```
 FORMATTER / Option to insert a space before comma in an allocation expression
     - option id:         "org.eclipse.jdt.core.formatter.insert_space_before_comma_in_allocation_expression"
     - possible values:   { INSERT, DO_NOT_INSERT }
     - default:           DO_NOT_INSERT
```

Since:

3.0

See Also:

JavaCore.INSERT, JavaCore.DO_NOT_INSERT, Constant Field Values

FORMATTER_INSERT_SPACE_BEFORE_COMMA_IN_ANNOTATION

```
public static final String  FORMATTER_INSERT_SPACE_BEFORE_COMMA_IN_ANNOTATION
```

```
 FORMATTER / Option to insert a space before comma in annotation
     - option id:         "org.eclipse.jdt.core.formatter.insert_space_before_comma_in_annotation"
     - possible values:   { INSERT, DO_NOT_INSERT }
     - default:           DO_NOT_INSERT
```

Since:

3.1

See Also:

JavaCore.INSERT, JavaCore.DO_NOT_INSERT, Constant Field Values

FORMATTER_INSERT_SPACE_BEFORE_COMMA_IN_ARRAY_INITIALIZER

```
public static final String  FORMATTER_INSERT_SPACE_BEFORE_COMMA_IN_ARRAY_INITIALIZER
```

```
 FORMATTER / Option to insert a space before comma in an array initializer
     - option id:         "org.eclipse.jdt.core.formatter.insert_space_before_comma_in_array_initializer"
     - possible values:   { INSERT, DO_NOT_INSERT }
     - default:           DO_NOT_INSERT
```

Since:

3.0

See Also:

JavaCore.INSERT, JavaCore.DO_NOT_INSERT, Constant Field Values

FORMATTER_INSERT_SPACE_BEFORE_COMMA_IN_CONSTRUCTOR_DECLARATION_PARAMETERS

```
public static final String  FORMATTER_INSERT_SPACE_BEFORE_COMMA_IN_CONSTRUCTOR_DECLARATION_PARAMETERS
```

```
 FORMATTER / Option to insert a space before comma in the parameters of a constructor declaration
     - option id:         "org.eclipse.jdt.core.formatter.insert_space_before_comma_in_constructor_declaration_parameters"
     - possible values:   { INSERT, DO_NOT_INSERT }
     - default:           DO_NOT_INSERT
```

Since:

3.0

See Also:

JavaCore.INSERT, JavaCore.DO_NOT_INSERT, Constant Field Values

FORMATTER_INSERT_SPACE_BEFORE_COMMA_IN_CONSTRUCTOR_DECLARATION_THROWS

```
public static final String  FORMATTER_INSERT_SPACE_BEFORE_COMMA_IN_CONSTRUCTOR_DECLARATION_THROWS
```

```
 FORMATTER / Option to insert a space before comma in the exception names of the throws clause of a constructor declaration
     - option id:         "org.eclipse.jdt.core.formatter.insert_space_before_comma_in_constructor_declaration_throws"
     - possible values:   { INSERT, DO_NOT_INSERT }
     - default:           DO_NOT_INSERT
```

Since:

3.0

See Also:

JavaCore.INSERT, JavaCore.DO_NOT_INSERT, Constant Field Values

FORMATTER_INSERT_SPACE_BEFORE_COMMA_IN_ENUM_CONSTANT_ARGUMENTS

```
public static final String  FORMATTER_INSERT_SPACE_BEFORE_COMMA_IN_ENUM_CONSTANT_ARGUMENTS
```

```
 FORMATTER / Option to insert a space before comma in the arguments of enum constant
     - option id:         "org.eclipse.jdt.core.formatter.insert_space_before_comma_in_enum_constant_arguments"
     - possible values:   { INSERT, DO_NOT_INSERT }
     - default:           DO_NOT_INSERT
```

Since:

3.1

See Also:

JavaCore.INSERT, JavaCore.DO_NOT_INSERT, Constant Field Values

FORMATTER_INSERT_SPACE_BEFORE_COMMA_IN_ENUM_DECLARATIONS

```
public static final String  FORMATTER_INSERT_SPACE_BEFORE_COMMA_IN_ENUM_DECLARATIONS
```

```
 FORMATTER / Option to insert a space before comma in enum declarations
     - option id:         "org.eclipse.jdt.core.formatter.insert_space_before_comma_in_enum_declarations"
     - possible values:   { INSERT, DO_NOT_INSERT }
     - default:           DO_NOT_INSERT
```

Since:

3.1

See Also:

JavaCore.INSERT, JavaCore.DO_NOT_INSERT, Constant Field Values

FORMATTER_INSERT_SPACE_BEFORE_COMMA_IN_EXPLICIT_CONSTRUCTOR_CALL_ARGUMENTS

```
public static final String  FORMATTER_INSERT_SPACE_BEFORE_COMMA_IN_EXPLICIT_CONSTRUCTOR_CALL_ARGUMENTS
```

```
 FORMATTER / Option to insert a space before comma in the arguments of an explicit constructor call
     - option id:         "org.eclipse.jdt.core.formatter.insert_space_before_comma_in_explicitconstructorcall_arguments"
     - possible values:   { INSERT, DO_NOT_INSERT }
     - default:           DO_NOT_INSERT
```

Since:

3.0

See Also:

JavaCore.INSERT, JavaCore.DO_NOT_INSERT, Constant Field Values

FORMATTER_INSERT_SPACE_BEFORE_COMMA_IN_FOR_INCREMENTS

```
public static final String  FORMATTER_INSERT_SPACE_BEFORE_COMMA_IN_FOR_INCREMENTS
```

```
 FORMATTER / Option to insert a space before comma in the increments of a for statement
     - option id:         "org.eclipse.jdt.core.formatter.insert_space_before_comma_in_for_increments"
     - possible values:   { INSERT, DO_NOT_INSERT }
     - default:           DO_NOT_INSERT
```

Since:

3.0

See Also:

JavaCore.INSERT, JavaCore.DO_NOT_INSERT, Constant Field Values

FORMATTER_INSERT_SPACE_BEFORE_COMMA_IN_FOR_INITS

```
public static final String  FORMATTER_INSERT_SPACE_BEFORE_COMMA_IN_FOR_INITS
```

```
 FORMATTER / Option to insert a space before comma in the initializations of a for statement
     - option id:         "org.eclipse.jdt.core.formatter.insert_space_before_comma_in_for_inits"
     - possible values:   { INSERT, DO_NOT_INSERT }
     - default:           DO_NOT_INSERT
```

Since:

3.0

See Also:

JavaCore.INSERT, JavaCore.DO_NOT_INSERT, Constant Field Values

FORMATTER_INSERT_SPACE_BEFORE_COMMA_IN_METHOD_DECLARATION_PARAMETERS

```
public static final String  FORMATTER_INSERT_SPACE_BEFORE_COMMA_IN_METHOD_DECLARATION_PARAMETERS
```

```
 FORMATTER / Option to insert a space before comma in the parameters of a method declaration
     - option id:         "org.eclipse.jdt.core.formatter.insert_space_before_comma_in_method_declaration_parameters"
     - possible values:   { INSERT, DO_NOT_INSERT }
     - default:           DO_NOT_INSERT
```

Since:

3.0

See Also:

JavaCore.INSERT, JavaCore.DO_NOT_INSERT, Constant Field Values

FORMATTER_INSERT_SPACE_BEFORE_COMMA_IN_METHOD_DECLARATION_THROWS

```
public static final String  FORMATTER_INSERT_SPACE_BEFORE_COMMA_IN_METHOD_DECLARATION_THROWS
```

```
 FORMATTER / Option to insert a space before comma in the exception names of the throws clause of a method declaration
     - option id:         "org.eclipse.jdt.core.formatter.insert_space_before_comma_in_method_declaration_throws"
     - possible values:   { INSERT, DO_NOT_INSERT }
     - default:           DO_NOT_INSERT
```

Since:

3.0

See Also:

JavaCore.INSERT, JavaCore.DO_NOT_INSERT, Constant Field Values

FORMATTER_INSERT_SPACE_BEFORE_COMMA_IN_METHOD_INVOCATION_ARGUMENTS

```
public static final String  FORMATTER_INSERT_SPACE_BEFORE_COMMA_IN_METHOD_INVOCATION_ARGUMENTS
```

```
 FORMATTER / Option to insert a space before comma in the arguments of a method invocation
     - option id:         "org.eclipse.jdt.core.formatter.insert_space_before_comma_in_method_invocation_arguments"
     - possible values:   { INSERT, DO_NOT_INSERT }
     - default:           DO_NOT_INSERT
```

Since:

3.0

See Also:

JavaCore.INSERT, JavaCore.DO_NOT_INSERT, Constant Field Values

FORMATTER_INSERT_SPACE_BEFORE_COMMA_IN_MULTIPLE_FIELD_DECLARATIONS

```
public static final String  FORMATTER_INSERT_SPACE_BEFORE_COMMA_IN_MULTIPLE_FIELD_DECLARATIONS
```

```
 FORMATTER / Option to insert a space before comma in a multiple field declaration
     - option id:         "org.eclipse.jdt.core.formatter.insert_space_before_comma_in_multiple_field_declarations"
     - possible values:   { INSERT, DO_NOT_INSERT }
     - default:           DO_NOT_INSERT
```

Since:

3.0

See Also:

JavaCore.INSERT, JavaCore.DO_NOT_INSERT, Constant Field Values

FORMATTER_INSERT_SPACE_BEFORE_COMMA_IN_MULTIPLE_LOCAL_DECLARATIONS

```
public static final String  FORMATTER_INSERT_SPACE_BEFORE_COMMA_IN_MULTIPLE_LOCAL_DECLARATIONS
```

```
 FORMATTER / Option to insert a space before comma in a multiple local declaration
     - option id:         "org.eclipse.jdt.core.formatter.insert_space_before_comma_in_multiple_local_declarations"
     - possible values:   { INSERT, DO_NOT_INSERT }
     - default:           DO_NOT_INSERT
```

Since:

3.0

See Also:

JavaCore.INSERT, JavaCore.DO_NOT_INSERT, Constant Field Values

FORMATTER_INSERT_SPACE_BEFORE_COMMA_IN_PARAMETERIZED_TYPE_REFERENCE

```
public static final String  FORMATTER_INSERT_SPACE_BEFORE_COMMA_IN_PARAMETERIZED_TYPE_REFERENCE
```

```
 FORMATTER / Option to insert a space before comma in parameterized type reference
     - option id:         "org.eclipse.jdt.core.formatter.insert_space_before_comma_in_parameterized_type_reference"
     - possible values:   { INSERT, DO_NOT_INSERT }
     - default:           DO_NOT_INSERT
```

Since:

3.1

See Also:

JavaCore.INSERT, JavaCore.DO_NOT_INSERT, Constant Field Values

FORMATTER_INSERT_SPACE_BEFORE_COMMA_IN_PERMITTED_TYPES

```
public static final String  FORMATTER_INSERT_SPACE_BEFORE_COMMA_IN_PERMITTED_TYPES
```

```
 FORMATTER / Option to insert a space before comma in the permitted types list in a type header
     - option id:         "org.eclipse.jdt.core.formatter.insert_space_before_comma_in_permitted_types"
     - possible values:   { INSERT, DO_NOT_INSERT }
     - default:           DO_NOT_INSERT
```

Since:

3.29

See Also:

JavaCore.INSERT, JavaCore.DO_NOT_INSERT, Constant Field Values

FORMATTER_INSERT_SPACE_BEFORE_COMMA_IN_RECORD_COMPONENTS

```
public static final String  FORMATTER_INSERT_SPACE_BEFORE_COMMA_IN_RECORD_COMPONENTS
```

```
 FORMATTER / Option to insert a space before comma in record components list
     - option id:         "org.eclipse.jdt.core.formatter.insert_space_before_comma_in_record_components"
     - possible values:   { INSERT, DO_NOT_INSERT }
     - default:           DO_NOT_INSERT
```

Since:

3.22

See Also:

JavaCore.INSERT, JavaCore.DO_NOT_INSERT, Constant Field Values

FORMATTER_INSERT_SPACE_BEFORE_COMMA_IN_SUPERINTERFACES

```
public static final String  FORMATTER_INSERT_SPACE_BEFORE_COMMA_IN_SUPERINTERFACES
```

```
 FORMATTER / Option to insert a space before comma in the superinterfaces names in a type header
     - option id:         "org.eclipse.jdt.core.formatter.insert_space_before_comma_in_superinterfaces"
     - possible values:   { INSERT, DO_NOT_INSERT }
     - default:           DO_NOT_INSERT
```

Since:

3.0

See Also:

JavaCore.INSERT, JavaCore.DO_NOT_INSERT, Constant Field Values

FORMATTER_INSERT_SPACE_BEFORE_COMMA_IN_SWITCH_CASE_EXPRESSIONS

```
public static final String  FORMATTER_INSERT_SPACE_BEFORE_COMMA_IN_SWITCH_CASE_EXPRESSIONS
```

```
 FORMATTER / Option to insert a space before the comma in switch case expressions list
     - option id:         "org.eclipse.jdt.core.formatter.insert_space_before_comma_in_switch_case_expressions"
     - possible values:   { INSERT, DO_NOT_INSERT }
     - default:           DO_NOT_INSERT
```

Since:

3.18

See Also:

JavaCore.INSERT, JavaCore.DO_NOT_INSERT, Constant Field Values

FORMATTER_INSERT_SPACE_BEFORE_COMMA_IN_TYPE_ARGUMENTS

```
public static final String  FORMATTER_INSERT_SPACE_BEFORE_COMMA_IN_TYPE_ARGUMENTS
```

```
 FORMATTER / Option to insert a space before comma in explicit type arguments on method/constructor invocations
     - option id:         "org.eclipse.jdt.core.formatter.insert_space_before_comma_in_type_arguments"
     - possible values:   { INSERT, DO_NOT_INSERT }
     - default:           DO_NOT_INSERT
```

Since:

3.1

See Also:

JavaCore.INSERT, JavaCore.DO_NOT_INSERT, Constant Field Values

FORMATTER_INSERT_SPACE_BEFORE_COMMA_IN_TYPE_PARAMETERS

```
public static final String  FORMATTER_INSERT_SPACE_BEFORE_COMMA_IN_TYPE_PARAMETERS
```

```
 FORMATTER / Option to insert a space before comma in type parameter declarations
     - option id:         "org.eclipse.jdt.core.formatter.insert_space_before_comma_in_type_parameters"
     - possible values:   { INSERT, DO_NOT_INSERT }
     - default:           DO_NOT_INSERT
```

Since:

3.1

See Also:

JavaCore.INSERT, JavaCore.DO_NOT_INSERT, Constant Field Values

FORMATTER_INSERT_SPACE_BEFORE_ELLIPSIS

```
public static final String  FORMATTER_INSERT_SPACE_BEFORE_ELLIPSIS
```

```
 FORMATTER / Option to insert a space before ellipsis
     - option id:         "org.eclipse.jdt.core.formatter.insert_space_before_ellipsis"
     - possible values:   { INSERT, DO_NOT_INSERT }
     - default:           DO_NOT_INSERT
```

Since:

3.1

See Also:

JavaCore.INSERT, JavaCore.DO_NOT_INSERT, Constant Field Values

FORMATTER_INSERT_SPACE_BEFORE_LAMBDA_ARROW

```
public static final String  FORMATTER_INSERT_SPACE_BEFORE_LAMBDA_ARROW
```

```
 FORMATTER / Option to insert a space before lambda ->
     - option id:         "org.eclipse.jdt.core.formatter.insert_space_before_lambda_arrow"
     - possible values:   { INSERT, DO_NOT_INSERT }
     - default:           INSERT
```

Since:

3.10

See Also:

JavaCore.INSERT, JavaCore.DO_NOT_INSERT, Constant Field Values

FORMATTER_INSERT_SPACE_BEFORE_OPENING_ANGLE_BRACKET_IN_PARAMETERIZED_TYPE_REFERENCE

```
public static final String  FORMATTER_INSERT_SPACE_BEFORE_OPENING_ANGLE_BRACKET_IN_PARAMETERIZED_TYPE_REFERENCE
```

```
 FORMATTER / Option to insert a space before the opening angle bracket in parameterized type reference
     - option id:         "org.eclipse.jdt.core.formatter.insert_space_before_opening_angle_bracket_in_parameterized_type_reference"
     - possible values:   { INSERT, DO_NOT_INSERT }
     - default:           DO_NOT_INSERT
```

Since:

3.1

See Also:

JavaCore.INSERT, JavaCore.DO_NOT_INSERT, Constant Field Values

FORMATTER_INSERT_SPACE_BEFORE_OPENING_ANGLE_BRACKET_IN_TYPE_ARGUMENTS

```
public static final String  FORMATTER_INSERT_SPACE_BEFORE_OPENING_ANGLE_BRACKET_IN_TYPE_ARGUMENTS
```

```
 FORMATTER / Option to insert a space before the opening angle bracket in explicit type arguments on method/constructor invocations
     - option id:         "org.eclipse.jdt.core.formatter.insert_space_before_opening_angle_bracket_in_type_arguments"
     - possible values:   { INSERT, DO_NOT_INSERT }
     - default:           DO_NOT_INSERT
```

Since:

3.1

See Also:

JavaCore.INSERT, JavaCore.DO_NOT_INSERT, Constant Field Values

FORMATTER_INSERT_SPACE_BEFORE_OPENING_ANGLE_BRACKET_IN_TYPE_PARAMETERS

```
public static final String  FORMATTER_INSERT_SPACE_BEFORE_OPENING_ANGLE_BRACKET_IN_TYPE_PARAMETERS
```

```
 FORMATTER / Option to insert a space before the opening angle bracket in type parameter declarations
     - option id:         "org.eclipse.jdt.core.formatter.insert_space_before_opening_angle_bracket_in_type_parameters"
     - possible values:   { INSERT, DO_NOT_INSERT }
     - default:           DO_NOT_INSERT
```

Since:

3.1

See Also:

JavaCore.INSERT, JavaCore.DO_NOT_INSERT, Constant Field Values

FORMATTER_INSERT_SPACE_BEFORE_OPENING_BRACE_IN_ANNOTATION_TYPE_DECLARATION

```
public static final String  FORMATTER_INSERT_SPACE_BEFORE_OPENING_BRACE_IN_ANNOTATION_TYPE_DECLARATION
```

```
 FORMATTER / Option to insert a space before the opening brace in an annotation type declaration
     - option id:         "org.eclipse.jdt.core.formatter.insert_space_before_opening_brace_in_annotation_type_declaration"
     - possible values:   { INSERT, DO_NOT_INSERT }
     - default:           INSERT
```

Since:

3.1

See Also:

JavaCore.INSERT, JavaCore.DO_NOT_INSERT, Constant Field Values

FORMATTER_INSERT_SPACE_BEFORE_OPENING_BRACE_IN_ANONYMOUS_TYPE_DECLARATION

```
public static final String  FORMATTER_INSERT_SPACE_BEFORE_OPENING_BRACE_IN_ANONYMOUS_TYPE_DECLARATION
```

```
 FORMATTER / Option to insert a space before the opening brace in an anonymous type declaration
     - option id:         "org.eclipse.jdt.core.formatter.insert_space_before_opening_brace_in_anonymous_type_declaration"
     - possible values:   { INSERT, DO_NOT_INSERT }
     - default:           INSERT
```

Since:

3.0

See Also:

JavaCore.INSERT, JavaCore.DO_NOT_INSERT, Constant Field Values

FORMATTER_INSERT_SPACE_BEFORE_OPENING_BRACE_IN_ARRAY_INITIALIZER

```
public static final String  FORMATTER_INSERT_SPACE_BEFORE_OPENING_BRACE_IN_ARRAY_INITIALIZER
```

```
 FORMATTER / Option to insert a space before the opening brace in an array initializer
     - option id:         "org.eclipse.jdt.core.formatter.insert_space_before_opening_brace_in_array_initializer"
     - possible values:   { INSERT, DO_NOT_INSERT }
     - default:           DO_NOT_INSERT
```

Since:

3.0

See Also:

JavaCore.INSERT, JavaCore.DO_NOT_INSERT, Constant Field Values

FORMATTER_INSERT_SPACE_BEFORE_OPENING_BRACE_IN_BLOCK

```
public static final String  FORMATTER_INSERT_SPACE_BEFORE_OPENING_BRACE_IN_BLOCK
```

```
 FORMATTER / Option to insert a space before the opening brace in a block
     - option id:         "org.eclipse.jdt.core.formatter.insert_space_before_opening_brace_in_block"
     - possible values:   { INSERT, DO_NOT_INSERT }
     - default:           INSERT
```

Since:

3.0

See Also:

JavaCore.INSERT, JavaCore.DO_NOT_INSERT, Constant Field Values

FORMATTER_INSERT_SPACE_BEFORE_OPENING_BRACE_IN_CONSTRUCTOR_DECLARATION

```
public static final String  FORMATTER_INSERT_SPACE_BEFORE_OPENING_BRACE_IN_CONSTRUCTOR_DECLARATION
```

```
 FORMATTER / Option to insert a space before the opening brace in a constructor declaration
     - option id:         "org.eclipse.jdt.core.formatter.insert_space_before_opening_brace_in_constructor_declaration"
     - possible values:   { INSERT, DO_NOT_INSERT }
     - default:           INSERT
```

Since:

3.0

See Also:

JavaCore.INSERT, JavaCore.DO_NOT_INSERT, Constant Field Values

FORMATTER_INSERT_SPACE_BEFORE_OPENING_BRACE_IN_ENUM_CONSTANT

```
public static final String  FORMATTER_INSERT_SPACE_BEFORE_OPENING_BRACE_IN_ENUM_CONSTANT
```

```
 FORMATTER / Option to insert a space before the opening brace in an enum constant
     - option id:         "org.eclipse.jdt.core.formatter.insert_space_before_opening_brace_in_enum_constant"
     - possible values:   { INSERT, DO_NOT_INSERT }
     - default:           INSERT
```

Since:

3.1

See Also:

JavaCore.INSERT, JavaCore.DO_NOT_INSERT, Constant Field Values

FORMATTER_INSERT_SPACE_BEFORE_OPENING_BRACE_IN_ENUM_DECLARATION

```
public static final String  FORMATTER_INSERT_SPACE_BEFORE_OPENING_BRACE_IN_ENUM_DECLARATION
```

```
 FORMATTER / Option to insert a space before the opening brace in an enum declaration
     - option id:         "org.eclipse.jdt.core.formatter.insert_space_before_opening_brace_in_enum_declaration"
     - possible values:   { INSERT, DO_NOT_INSERT }
     - default:           INSERT
```

Since:

3.1

See Also:

JavaCore.INSERT, JavaCore.DO_NOT_INSERT, Constant Field Values

FORMATTER_INSERT_SPACE_BEFORE_OPENING_BRACE_IN_METHOD_DECLARATION

```
public static final String  FORMATTER_INSERT_SPACE_BEFORE_OPENING_BRACE_IN_METHOD_DECLARATION
```

```
 FORMATTER / Option to insert a space before the opening brace in a method declaration
     - option id:         "org.eclipse.jdt.core.formatter.insert_space_before_opening_brace_in_method_declaration"
     - possible values:   { INSERT, DO_NOT_INSERT }
     - default:           INSERT
```

Since:

3.0

See Also:

JavaCore.INSERT, JavaCore.DO_NOT_INSERT, Constant Field Values

FORMATTER_INSERT_SPACE_BEFORE_OPENING_BRACE_IN_RECORD_CONSTRUCTOR

```
public static final String  FORMATTER_INSERT_SPACE_BEFORE_OPENING_BRACE_IN_RECORD_CONSTRUCTOR
```

```
 FORMATTER / Option to insert a space before the opening brace in a record constructor
     - option id:         "org.eclipse.jdt.core.formatter.insert_space_before_opening_brace_in_record_constructor"
     - possible values:   { INSERT, DO_NOT_INSERT }
     - default:           INSERT
```

Since:

3.22

See Also:

JavaCore.INSERT, JavaCore.DO_NOT_INSERT, Constant Field Values

FORMATTER_INSERT_SPACE_BEFORE_OPENING_BRACE_IN_RECORD_DECLARATION

```
public static final String  FORMATTER_INSERT_SPACE_BEFORE_OPENING_BRACE_IN_RECORD_DECLARATION
```

```
 FORMATTER / Option to insert a space before the opening brace in a record declaration
     - option id:         "org.eclipse.jdt.core.formatter.insert_space_before_opening_brace_in_record_declaration"
     - possible values:   { INSERT, DO_NOT_INSERT }
     - default:           INSERT
```

Since:

3.22

See Also:

JavaCore.INSERT, JavaCore.DO_NOT_INSERT, Constant Field Values

FORMATTER_INSERT_SPACE_BEFORE_OPENING_BRACE_IN_SWITCH

```
public static final String  FORMATTER_INSERT_SPACE_BEFORE_OPENING_BRACE_IN_SWITCH
```

```
 FORMATTER / Option to insert a space before the opening brace in a switch statement
     - option id:         "org.eclipse.jdt.core.formatter.insert_space_before_opening_brace_in_switch"
     - possible values:   { INSERT, DO_NOT_INSERT }
     - default:           INSERT
```

Since:

3.0

See Also:

JavaCore.INSERT, JavaCore.DO_NOT_INSERT, Constant Field Values

FORMATTER_INSERT_SPACE_BEFORE_OPENING_BRACE_IN_TYPE_DECLARATION

```
public static final String  FORMATTER_INSERT_SPACE_BEFORE_OPENING_BRACE_IN_TYPE_DECLARATION
```

```
 FORMATTER / Option to insert a space before the opening brace in a type declaration
     - option id:         "org.eclipse.jdt.core.formatter.insert_space_before_opening_brace_in_type_declaration"
     - possible values:   { INSERT, DO_NOT_INSERT }
     - default:           INSERT
```

Since:

3.0

See Also:

JavaCore.INSERT, JavaCore.DO_NOT_INSERT, Constant Field Values

FORMATTER_INSERT_SPACE_BEFORE_OPENING_BRACKET_IN_ARRAY_ALLOCATION_EXPRESSION

```
public static final String  FORMATTER_INSERT_SPACE_BEFORE_OPENING_BRACKET_IN_ARRAY_ALLOCATION_EXPRESSION
```

```
 FORMATTER / Option to insert a space before the opening bracket in an array allocation expression
     - option id:         "org.eclipse.jdt.core.formatter.insert_space_before_opening_bracket_in_array_allocation_expression"
     - possible values:   { INSERT, DO_NOT_INSERT }
     - default:           DO_NOT_INSERT
```

Since:

3.0

See Also:

JavaCore.INSERT, JavaCore.DO_NOT_INSERT, Constant Field Values

FORMATTER_INSERT_SPACE_BEFORE_OPENING_BRACKET_IN_ARRAY_REFERENCE

```
public static final String  FORMATTER_INSERT_SPACE_BEFORE_OPENING_BRACKET_IN_ARRAY_REFERENCE
```

```
 FORMATTER / Option to insert a space before the opening bracket in an array reference
     - option id:         "org.eclipse.jdt.core.formatter.insert_space_before_opening_bracket_in_array_reference"
     - possible values:   { INSERT, DO_NOT_INSERT }
     - default:           DO_NOT_INSERT
```

Since:

3.0

See Also:

JavaCore.INSERT, JavaCore.DO_NOT_INSERT, Constant Field Values

FORMATTER_INSERT_SPACE_BEFORE_OPENING_BRACKET_IN_ARRAY_TYPE_REFERENCE

```
public static final String  FORMATTER_INSERT_SPACE_BEFORE_OPENING_BRACKET_IN_ARRAY_TYPE_REFERENCE
```

```
 FORMATTER / Option to insert a space before the opening bracket in an array type reference
     - option id:         "org.eclipse.jdt.core.formatter.insert_space_before_opening_bracket_in_array_type_reference"
     - possible values:   { INSERT, DO_NOT_INSERT }
     - default:           DO_NOT_INSERT
```

Since:

3.0

See Also:

JavaCore.INSERT, JavaCore.DO_NOT_INSERT, Constant Field Values

FORMATTER_INSERT_SPACE_BEFORE_OPENING_PAREN_IN_ANNOTATION

```
public static final String  FORMATTER_INSERT_SPACE_BEFORE_OPENING_PAREN_IN_ANNOTATION
```

```
 FORMATTER / Option to insert a space before the opening parenthesis in annotation
     - option id:         "org.eclipse.jdt.core.formatter.insert_space_before_opening_paren_in_annotation"
     - possible values:   { INSERT, DO_NOT_INSERT }
     - default:           DO_NOT_INSERT
```

Since:

3.1

See Also:

JavaCore.INSERT, JavaCore.DO_NOT_INSERT, Constant Field Values

FORMATTER_INSERT_SPACE_BEFORE_OPENING_PAREN_IN_ANNOTATION_TYPE_MEMBER_DECLARATION

```
public static final String  FORMATTER_INSERT_SPACE_BEFORE_OPENING_PAREN_IN_ANNOTATION_TYPE_MEMBER_DECLARATION
```

```
 FORMATTER / Option to insert a space before the opening parenthesis in annotation type member declaration
     - option id:         "org.eclipse.jdt.core.formatter.insert_space_before_opening_paren_in_annotation_type_member_declaration"
     - possible values:   { INSERT, DO_NOT_INSERT }
     - default:           DO_NOT_INSERT
```

Since:

3.1

See Also:

JavaCore.INSERT, JavaCore.DO_NOT_INSERT, Constant Field Values

FORMATTER_INSERT_SPACE_BEFORE_OPENING_PAREN_IN_CATCH

```
public static final String  FORMATTER_INSERT_SPACE_BEFORE_OPENING_PAREN_IN_CATCH
```

```
 FORMATTER / Option to insert a space before the opening parenthesis in a catch
     - option id:         "org.eclipse.jdt.core.formatter.insert_space_before_opening_paren_in_catch"
     - possible values:   { INSERT, DO_NOT_INSERT }
     - default:           INSERT
```

Since:

3.0

See Also:

JavaCore.INSERT, JavaCore.DO_NOT_INSERT, Constant Field Values

FORMATTER_INSERT_SPACE_BEFORE_OPENING_PAREN_IN_CONSTRUCTOR_DECLARATION

```
public static final String  FORMATTER_INSERT_SPACE_BEFORE_OPENING_PAREN_IN_CONSTRUCTOR_DECLARATION
```

```
 FORMATTER / Option to insert a space before the opening parenthesis in a constructor declaration
     - option id:         "org.eclipse.jdt.core.formatter.insert_space_before_opening_paren_in_constructor_declaration"
     - possible values:   { INSERT, DO_NOT_INSERT }
     - default:           DO_NOT_INSERT
```

Since:

3.0

See Also:

JavaCore.INSERT, JavaCore.DO_NOT_INSERT, Constant Field Values

FORMATTER_INSERT_SPACE_BEFORE_OPENING_PAREN_IN_ENUM_CONSTANT

```
public static final String  FORMATTER_INSERT_SPACE_BEFORE_OPENING_PAREN_IN_ENUM_CONSTANT
```

```
 FORMATTER / Option to insert a space before the opening parenthesis in enum constant
     - option id:         "org.eclipse.jdt.core.formatter.insert_space_before_opening_paren_in_enum_constant"
     - possible values:   { INSERT, DO_NOT_INSERT }
     - default:           DO_NOT_INSERT
```

Since:

3.1

See Also:

JavaCore.INSERT, JavaCore.DO_NOT_INSERT, Constant Field Values

FORMATTER_INSERT_SPACE_BEFORE_OPENING_PAREN_IN_FOR

```
public static final String  FORMATTER_INSERT_SPACE_BEFORE_OPENING_PAREN_IN_FOR
```

```
 FORMATTER / Option to insert a space before the opening parenthesis in a for statement
     - option id:         "org.eclipse.jdt.core.formatter.insert_space_before_opening_paren_in_for"
     - possible values:   { INSERT, DO_NOT_INSERT }
     - default:           INSERT
```

Since:

3.0

See Also:

JavaCore.INSERT, JavaCore.DO_NOT_INSERT, Constant Field Values

FORMATTER_INSERT_SPACE_BEFORE_OPENING_PAREN_IN_IF

```
public static final String  FORMATTER_INSERT_SPACE_BEFORE_OPENING_PAREN_IN_IF
```

```
 FORMATTER / Option to insert a space before the opening parenthesis in an if statement
     - option id:         "org.eclipse.jdt.core.formatter.insert_space_before_opening_paren_in_if"
     - possible values:   { INSERT, DO_NOT_INSERT }
     - default:           INSERT
```

Since:

3.0

See Also:

JavaCore.INSERT, JavaCore.DO_NOT_INSERT, Constant Field Values

FORMATTER_INSERT_SPACE_BEFORE_OPENING_PAREN_IN_METHOD_DECLARATION

```
public static final String  FORMATTER_INSERT_SPACE_BEFORE_OPENING_PAREN_IN_METHOD_DECLARATION
```

```
 FORMATTER / Option to insert a space before the opening parenthesis in a method declaration
     - option id:         "org.eclipse.jdt.core.formatter.insert_space_before_opening_paren_in_method_declaration"
     - possible values:   { INSERT, DO_NOT_INSERT }
     - default:           DO_NOT_INSERT
```

Since:

3.0

See Also:

JavaCore.INSERT, JavaCore.DO_NOT_INSERT, Constant Field Values

FORMATTER_INSERT_SPACE_BEFORE_OPENING_PAREN_IN_METHOD_INVOCATION

```
public static final String  FORMATTER_INSERT_SPACE_BEFORE_OPENING_PAREN_IN_METHOD_INVOCATION
```

```
 FORMATTER / Option to insert a space before the opening parenthesis in a method invocation
     - option id:         "org.eclipse.jdt.core.formatter.insert_space_before_opening_paren_in_method_invocation"
     - possible values:   { INSERT, DO_NOT_INSERT }
     - default:           DO_NOT_INSERT
```

Since:

3.0

See Also:

JavaCore.INSERT, JavaCore.DO_NOT_INSERT, Constant Field Values

FORMATTER_INSERT_SPACE_BEFORE_OPENING_PAREN_IN_PARENTHESIZED_EXPRESSION

```
public static final String  FORMATTER_INSERT_SPACE_BEFORE_OPENING_PAREN_IN_PARENTHESIZED_EXPRESSION
```

```
 FORMATTER / Option to insert a space before the opening parenthesis in a parenthesized expression
     - option id:         "org.eclipse.jdt.core.formatter.insert_space_before_opening_paren_in_parenthesized_expression"
     - possible values:   { INSERT, DO_NOT_INSERT }
     - default:           DO_NOT_INSERT
```

Since:

3.0

See Also:

JavaCore.INSERT, JavaCore.DO_NOT_INSERT, Constant Field Values

FORMATTER_INSERT_SPACE_BEFORE_OPENING_PAREN_IN_RECORD_DECLARATION

```
public static final String  FORMATTER_INSERT_SPACE_BEFORE_OPENING_PAREN_IN_RECORD_DECLARATION
```

```
 FORMATTER / Option to insert a space before the opening parenthesis in a record declaration
     - option id:         "org.eclipse.jdt.core.formatter.insert_space_before_opening_paren_in_record_declaration"
     - possible values:   { INSERT, DO_NOT_INSERT }
     - default:           DO_NOT_INSERT
```

Since:

3.22

See Also:

JavaCore.INSERT, JavaCore.DO_NOT_INSERT, Constant Field Values

FORMATTER_INSERT_SPACE_BEFORE_OPENING_PAREN_IN_SWITCH

```
public static final String  FORMATTER_INSERT_SPACE_BEFORE_OPENING_PAREN_IN_SWITCH
```

```
 FORMATTER / Option to insert a space before the opening parenthesis in a switch statement
     - option id:         "org.eclipse.jdt.core.formatter.insert_space_before_opening_paren_in_switch"
     - possible values:   { INSERT, DO_NOT_INSERT }
     - default:           INSERT
```

Since:

3.0

See Also:

JavaCore.INSERT, JavaCore.DO_NOT_INSERT, Constant Field Values

FORMATTER_INSERT_SPACE_BEFORE_OPENING_PAREN_IN_SYNCHRONIZED

```
public static final String  FORMATTER_INSERT_SPACE_BEFORE_OPENING_PAREN_IN_SYNCHRONIZED
```

```
 FORMATTER / Option to insert a space before the opening parenthesis in a synchronized statement
     - option id:         "org.eclipse.jdt.core.formatter.insert_space_before_opening_paren_in_synchronized"
     - possible values:   { INSERT, DO_NOT_INSERT }
     - default:           INSERT
```

Since:

3.0

See Also:

JavaCore.INSERT, JavaCore.DO_NOT_INSERT, Constant Field Values

FORMATTER_INSERT_SPACE_BEFORE_OPENING_PAREN_IN_TRY

```
public static final String  FORMATTER_INSERT_SPACE_BEFORE_OPENING_PAREN_IN_TRY
```

```
 FORMATTER / Option to insert a space before the opening parenthesis in a try with resources statement
     - option id:         "org.eclipse.jdt.core.formatter.insert_space_before_opening_paren_in_try"
     - possible values:   { INSERT, DO_NOT_INSERT }
     - default:           INSERT
```

Since:

3.7.1

See Also:

JavaCore.INSERT, JavaCore.DO_NOT_INSERT, Constant Field Values

FORMATTER_INSERT_SPACE_BEFORE_OPENING_PAREN_IN_WHILE

```
public static final String  FORMATTER_INSERT_SPACE_BEFORE_OPENING_PAREN_IN_WHILE
```

```
 FORMATTER / Option to insert a space before the opening parenthesis in a while statement
     - option id:         "org.eclipse.jdt.core.formatter.insert_space_before_opening_paren_in_while"
     - possible values:   { INSERT, DO_NOT_INSERT }
     - default:           INSERT
```

Since:

3.0

See Also:

JavaCore.INSERT, JavaCore.DO_NOT_INSERT, Constant Field Values

FORMATTER_INSERT_SPACE_BEFORE_PARENTHESIZED_EXPRESSION_IN_RETURN

```
public static final String  FORMATTER_INSERT_SPACE_BEFORE_PARENTHESIZED_EXPRESSION_IN_RETURN
```

```
 FORMATTER / Option to insert a space before parenthesized expression in return statement
     - option id:         "org.eclipse.jdt.core.formatter.insert_space_before_parenthesized_expression_in_return"
     - possible values:   { INSERT, DO_NOT_INSERT }
     - default:           INSERT
```

Since:

3.2

See Also:

JavaCore.INSERT, JavaCore.DO_NOT_INSERT, Constant Field Values

FORMATTER_INSERT_SPACE_BEFORE_PARENTHESIZED_EXPRESSION_IN_THROW

```
public static final String  FORMATTER_INSERT_SPACE_BEFORE_PARENTHESIZED_EXPRESSION_IN_THROW
```

```
 FORMATTER / Option to insert a space before parenthesized expression in throw statement
     - option id:         "org.eclipse.jdt.core.formatter.insert_space_before_parenthesized_expression_in_throw"
     - possible values:   { INSERT, DO_NOT_INSERT }
     - default:           INSERT
```

Since:

3.3

See Also:

JavaCore.INSERT, JavaCore.DO_NOT_INSERT, Constant Field Values

FORMATTER_INSERT_SPACE_BEFORE_POSTFIX_OPERATOR

```
public static final String  FORMATTER_INSERT_SPACE_BEFORE_POSTFIX_OPERATOR
```

```
 FORMATTER / Option to insert a space before a postfix operator
     - option id:         "org.eclipse.jdt.core.formatter.insert_space_before_postfix_operator"
     - possible values:   { INSERT, DO_NOT_INSERT }
     - default:           DO_NOT_INSERT
```

Since:

3.0

See Also:

JavaCore.INSERT, JavaCore.DO_NOT_INSERT, Constant Field Values

FORMATTER_INSERT_SPACE_BEFORE_PREFIX_OPERATOR

```
public static final String  FORMATTER_INSERT_SPACE_BEFORE_PREFIX_OPERATOR
```

```
 FORMATTER / Option to insert a space before a prefix operator
     - option id:         "org.eclipse.jdt.core.formatter.insert_space_before_prefix_operator"
     - possible values:   { INSERT, DO_NOT_INSERT }
     - default:           DO_NOT_INSERT
```

Since:

3.0

See Also:

JavaCore.INSERT, JavaCore.DO_NOT_INSERT, Constant Field Values

FORMATTER_INSERT_SPACE_BEFORE_QUESTION_IN_CONDITIONAL

```
public static final String  FORMATTER_INSERT_SPACE_BEFORE_QUESTION_IN_CONDITIONAL
```

```
 FORMATTER / Option to insert a space before question mark in a conditional expression
     - option id:         "org.eclipse.jdt.core.formatter.insert_space_before_question_in_conditional"
     - possible values:   { INSERT, DO_NOT_INSERT }
     - default:           INSERT
```

Since:

3.0

See Also:

JavaCore.INSERT, JavaCore.DO_NOT_INSERT, Constant Field Values

FORMATTER_INSERT_SPACE_BEFORE_QUESTION_IN_WILDCARD

```
public static final String  FORMATTER_INSERT_SPACE_BEFORE_QUESTION_IN_WILDCARD
```

```
 FORMATTER / Option to insert a space before question mark in a wildcard
     - option id:         "org.eclipse.jdt.core.formatter.insert_space_before_question_in_wildcard"
     - possible values:   { INSERT, DO_NOT_INSERT }
     - default:           DO_NOT_INSERT
```

Since:

3.1

See Also:

JavaCore.INSERT, JavaCore.DO_NOT_INSERT, Constant Field Values

FORMATTER_INSERT_SPACE_BEFORE_SEMICOLON

```
public static final String  FORMATTER_INSERT_SPACE_BEFORE_SEMICOLON
```

```
 FORMATTER / Option to insert a space before semicolon
     - option id:         "org.eclipse.jdt.core.formatter.insert_space_before_semicolon"
     - possible values:   { INSERT, DO_NOT_INSERT }
     - default:           DO_NOT_INSERT
```

Since:

3.0

See Also:

JavaCore.INSERT, JavaCore.DO_NOT_INSERT, Constant Field Values

FORMATTER_INSERT_SPACE_BEFORE_SEMICOLON_IN_FOR

```
public static final String  FORMATTER_INSERT_SPACE_BEFORE_SEMICOLON_IN_FOR
```

```
 FORMATTER / Option to insert a space before semicolon in for statement
     - option id:         "org.eclipse.jdt.core.formatter.insert_space_before_semicolon_in_for"
     - possible values:   { INSERT, DO_NOT_INSERT }
     - default:           DO_NOT_INSERT
```

Since:

3.0

See Also:

JavaCore.INSERT, JavaCore.DO_NOT_INSERT, Constant Field Values

FORMATTER_INSERT_SPACE_BEFORE_SEMICOLON_IN_TRY_RESOURCES

```
public static final String  FORMATTER_INSERT_SPACE_BEFORE_SEMICOLON_IN_TRY_RESOURCES
```

```
 FORMATTER / Option to insert a space before semicolons following each resource declaration in a try with
 resources statement
     - option id:         "org.eclipse.jdt.core.formatter.insert_space_before_semicolon_in_try_resources"
     - possible values:   { INSERT, DO_NOT_INSERT }
```

```
     - default:          DO_NOT_INSERT
```

Since:

3.7.1

See Also:

JavaCore.INSERT, JavaCore.DO_NOT_INSERT, Constant Field Values

FORMATTER_INSERT_SPACE_BEFORE_UNARY_OPERATOR

```
public static final String  FORMATTER_INSERT_SPACE_BEFORE_UNARY_OPERATOR
```

```
 FORMATTER / Option to insert a space before unary operator
     - option id:         "org.eclipse.jdt.core.formatter.insert_space_before_unary_operator"
     - possible values:   { INSERT, DO_NOT_INSERT }
     - default:          DO_NOT_INSERT
```

Since:

3.0

See Also:

JavaCore.INSERT, JavaCore.DO_NOT_INSERT, Constant Field Values

FORMATTER_INSERT_SPACE_BETWEEN_BRACKETS_IN_ARRAY_TYPE_REFERENCE

```
public static final String  FORMATTER_INSERT_SPACE_BETWEEN_BRACKETS_IN_ARRAY_TYPE_REFERENCE
```

```
 FORMATTER / Option to insert a space between brackets in an array type reference
     - option id:         "org.eclipse.jdt.core.formatter.insert_space_between_brackets_in_array_type_reference"
     - possible values:   { INSERT, DO_NOT_INSERT }
     - default:          DO_NOT_INSERT
```

Since:

3.0

See Also:

JavaCore.INSERT, JavaCore.DO_NOT_INSERT, Constant Field Values

FORMATTER_INSERT_SPACE_BETWEEN_EMPTY_BRACES_IN_ARRAY_INITIALIZER

```
public static final String  FORMATTER_INSERT_SPACE_BETWEEN_EMPTY_BRACES_IN_ARRAY_INITIALIZER
```

```
 FORMATTER / Option to insert a space between empty braces in an array initializer
     - option id:         "org.eclipse.jdt.core.formatter.insert_space_between_empty_braces_in_array_initializer"
     - possible values:   { INSERT, DO_NOT_INSERT }
     - default:          DO_NOT_INSERT
```

Since:

3.0

See Also:

JavaCore.INSERT, JavaCore.DO_NOT_INSERT, Constant Field Values

FORMATTER_INSERT_SPACE_BETWEEN_EMPTY_BRACKETS_IN_ARRAY_ALLOCATION_EXPRESSION

```
public static final String  FORMATTER_INSERT_SPACE_BETWEEN_EMPTY_BRACKETS_IN_ARRAY_ALLOCATION_EXPRESSION
```

```
 FORMATTER / Option to insert a space between empty brackets in an array allocation expression
     - option id:         "org.eclipse.jdt.core.formatter.insert_space_between_empty_brackets_in_array_allocation_expression"
     - possible values:   { INSERT, DO_NOT_INSERT }
     - default:          DO_NOT_INSERT
```

Since:

3.0

See Also:

JavaCore.INSERT, JavaCore.DO_NOT_INSERT, Constant Field Values

FORMATTER_INSERT_SPACE_BETWEEN_EMPTY_PARENS_IN_ANNOTATION_TYPE_MEMBER_DECLARATION

```
public static final String  FORMATTER_INSERT_SPACE_BETWEEN_EMPTY_PARENS_IN_ANNOTATION_TYPE_MEMBER_DECLARATION
```

```
 FORMATTER / Option to insert a space between empty parenthesis in an annotation type member declaration
     - option id:         "org.eclipse.jdt.core.formatter.insert_space_between_empty_parens_in_annotation_type_member_declaration"
     - possible values:   { INSERT, DO_NOT_INSERT }
```

```
     - default:           DO_NOT_INSERT
```

Since:

3.1

See Also:

JavaCore.INSERT, JavaCore.DO_NOT_INSERT, Constant Field Values

FORMATTER_INSERT_SPACE_BETWEEN_EMPTY_PARENS_IN_CONSTRUCTOR_DECLARATION

```
public static final String  FORMATTER_INSERT_SPACE_BETWEEN_EMPTY_PARENS_IN_CONSTRUCTOR_DECLARATION
```

```
 FORMATTER / Option to insert a space between empty parenthesis in a constructor declaration
     - option id:         "org.eclipse.jdt.core.formatter.insert_space_between_empty_parens_in_constructor_declaration"
     - possible values:   { INSERT, DO_NOT_INSERT }
     - default:           DO_NOT_INSERT
```

Since:

3.0

See Also:

JavaCore.INSERT, JavaCore.DO_NOT_INSERT, Constant Field Values

FORMATTER_INSERT_SPACE_BETWEEN_EMPTY_PARENS_IN_ENUM_CONSTANT

```
public static final String  FORMATTER_INSERT_SPACE_BETWEEN_EMPTY_PARENS_IN_ENUM_CONSTANT
```

```
 FORMATTER / Option to insert a space between empty parenthesis in enum constant
     - option id:         "org.eclipse.jdt.core.formatter.insert_space_between_empty_parens_in_enum_constant"
     - possible values:   { INSERT, DO_NOT_INSERT }
     - default:           DO_NOT_INSERT
```

Since:

3.1

See Also:

JavaCore.INSERT, JavaCore.DO_NOT_INSERT, Constant Field Values

FORMATTER_INSERT_SPACE_BETWEEN_EMPTY_PARENS_IN_METHOD_DECLARATION

```
public static final String  FORMATTER_INSERT_SPACE_BETWEEN_EMPTY_PARENS_IN_METHOD_DECLARATION
```

```
 FORMATTER / Option to insert a space between empty parenthesis in a method declaration
     - option id:         "org.eclipse.jdt.core.formatter.insert_space_between_empty_parens_in_method_declaration"
     - possible values:   { INSERT, DO_NOT_INSERT }
     - default:           DO_NOT_INSERT
```

Since:

3.0

See Also:

JavaCore.INSERT, JavaCore.DO_NOT_INSERT, Constant Field Values

FORMATTER_INSERT_SPACE_BETWEEN_EMPTY_PARENS_IN_METHOD_INVOCATION

```
public static final String  FORMATTER_INSERT_SPACE_BETWEEN_EMPTY_PARENS_IN_METHOD_INVOCATION
```

```
 FORMATTER / Option to insert a space between empty parenthesis in a method invocation
     - option id:         "org.eclipse.jdt.core.formatter.insert_space_between_empty_parens_in_method_invocation"
     - possible values:   { INSERT, DO_NOT_INSERT }
     - default:           DO_NOT_INSERT
```

Since:

3.0

See Also:

JavaCore.INSERT, JavaCore.DO_NOT_INSERT, Constant Field Values

FORMATTER_KEEP_ELSE_STATEMENT_ON_SAME_LINE

```
public static final String  FORMATTER_KEEP_ELSE_STATEMENT_ON_SAME_LINE
```

```
 FORMATTER / Option to keep else statement on the same line
     - option id:         "org.eclipse.jdt.core.formatter.keep_else_statement_on_same_line"
     - possible values:   { TRUE, FALSE }
```

```
     - default:           FALSE
```

Since:

3.0

See Also:

TRUE, FALSE, Constant Field Values

FORMATTER_KEEP_EMPTY_ARRAY_INITIALIZER_ON_ONE_LINE

```
public static final String  FORMATTER_KEEP_EMPTY_ARRAY_INITIALIZER_ON_ONE_LINE
```

```
 FORMATTER / Option to keep empty array initializer one one line
     - option id:         "org.eclipse.jdt.core.formatter.keep_empty_array_initializer_on_one_line"
     - possible values:   { TRUE, FALSE }
     - default:           FALSE
```

Since:

3.0

See Also:

TRUE, FALSE, Constant Field Values

FORMATTER_KEEP_GUARDIAN_CLAUSE_ON_ONE_LINE

```
public static final String  FORMATTER_KEEP_GUARDIAN_CLAUSE_ON_ONE_LINE
```

```
 FORMATTER / Option to keep guardian clause on one line, in addition to the
             #FORMATTER_KEEP_IF_THEN_BODY_BLOCK_ON_ONE_LINE option
     - option id:         "org.eclipse.jdt.core.formatter.format_guardian_clause_on_one_line"
     - possible values:   { TRUE, FALSE }
     - default:           FALSE
```

Since:

3.0

See Also:

TRUE,
FALSE,
FORMATTER_KEEP_IF_THEN_BODY_BLOCK_ON_ONE_LINE,
Constant Field Values

FORMATTER_KEEP_SIMPLE_IF_ON_ONE_LINE

```
public static final String  FORMATTER_KEEP_SIMPLE_IF_ON_ONE_LINE
```

```
 FORMATTER / Option to keep simple if statement on the one line
     - option id:         "org.eclipse.jdt.core.formatter.keep_imple_if_on_one_line"
     - possible values:   { TRUE, FALSE }
     - default:           FALSE
```

Since:

3.0

See Also:

TRUE, FALSE, Constant Field Values

FORMATTER_KEEP_THEN_STATEMENT_ON_SAME_LINE

```
public static final String  FORMATTER_KEEP_THEN_STATEMENT_ON_SAME_LINE
```

```
 FORMATTER / Option to keep then statement on the same line
     - option id:         "org.eclipse.jdt.core.formatter.keep_then_statement_on_same_line"
     - possible values:   { TRUE, FALSE }
     - default:           FALSE
```

Since:

3.0

See Also:

TRUE, FALSE, Constant Field Values

FORMATTER_KEEP_SIMPLE_FOR_BODY_ON_SAME_LINE

```
public static final String  FORMATTER_KEEP_SIMPLE_FOR_BODY_ON_SAME_LINE
```

```
 FORMATTER / Option to keep a simple 'for' loop body on the same line
     - option id:         "org.eclipse.jdt.core.formatter.keep_simple_for_body_on_same_line"
     - possible values:   { TRUE, FALSE }
     - default:           FALSE
```

Since:

3.15

See Also:

TRUE, FALSE, Constant Field Values

FORMATTER_KEEP_SIMPLE_WHILE_BODY_ON_SAME_LINE

```
public static final String  FORMATTER_KEEP_SIMPLE_WHILE_BODY_ON_SAME_LINE
```

```
 FORMATTER / Option to keep a simple 'while' loop body on the same line
     - option id:         "org.eclipse.jdt.core.formatter.keep_simple_while_body_on_same_line"
     - possible values:   { TRUE, FALSE }
     - default:           FALSE
```

Since:

3.15

See Also:

TRUE, FALSE, Constant Field Values

FORMATTER_KEEP_SIMPLE_DO_WHILE_BODY_ON_SAME_LINE

```
public static final String  FORMATTER_KEEP_SIMPLE_DO_WHILE_BODY_ON_SAME_LINE
```

```
 FORMATTER / Option to keep a simple 'do-while' loop body on the same line
     - option id:         "org.eclipse.jdt.core.formatter.keep_simple_do_while_body_on_same_line"
     - possible values:   { TRUE, FALSE }
     - default:           FALSE
```

Since:

3.15

See Also:

TRUE, FALSE, Constant Field Values

FORMATTER_KEEP_LOOP_BODY_BLOCK_ON_ONE_LINE

```
public static final String  FORMATTER_KEEP_LOOP_BODY_BLOCK_ON_ONE_LINE
```

```
 FORMATTER / Option to control when a loop body block should be kept on one line
     - option id:         "org.eclipse.jdt.core.formatter.keep_loop_body_block_on_one_line"
     - possible values:   { ONE_LINE_NEVER, ONE_LINE_IF_EMPTY, ONE_LINE_IF_SINGLE_ITEM,
                            ONE_LINE_ALWAYS, ONE_LINE_PRESERVE }
     - default:           ONE_LINE_NEVER
```

Since:

3.16

See Also:

ONE_LINE_NEVER, ONE_LINE_IF_EMPTY, ONE_LINE_IF_SINGLE_ITEM, ONE_LINE_ALWAYS, ONE_LINE_PRESERVE, Constant Field Values

FORMATTER_KEEP_IF_THEN_BODY_BLOCK_ON_ONE_LINE

```
public static final String  FORMATTER_KEEP_IF_THEN_BODY_BLOCK_ON_ONE_LINE
```

```
 FORMATTER / Option to control when an if-then statement body block should be kept on one line
     - option id:         "org.eclipse.jdt.core.formatter.keep_if_then_body_block_on_one_line"
     - possible values:   { ONE_LINE_NEVER, ONE_LINE_IF_EMPTY, ONE_LINE_IF_SINGLE_ITEM,
                            ONE_LINE_ALWAYS, ONE_LINE_PRESERVE }
     - default:           ONE_LINE_NEVER
```

Since:

3.16

See Also:

ONE_LINE_NEVER, ONE_LINE_IF_EMPTY, ONE_LINE_IF_SINGLE_ITEM, ONE_LINE_ALWAYS, ONE_LINE_PRESERVE, for a special case, Constant Field Values

FORMATTER_KEEP_SWITCH_BODY_BLOCK_ON_ONE_LINE

```
public static final String  FORMATTER_KEEP_SWITCH_BODY_BLOCK_ON_ONE_LINE
```

```
 FORMATTER / Option to control when the body of a switch statement/expression with arrows should be kept on one line
     - option id:         "org.eclipse.jdt.core.formatter.keep_switch_body_block_on_one_line"
     - possible values:   { ONE_LINE_NEVER, ONE_LINE_IF_EMPTY, ONE_LINE_IF_SINGLE_ITEM,
                            ONE_LINE_ALWAYS, ONE_LINE_PRESERVE }
     - default:           ONE_LINE_NEVER
```

Since:

3.29

See Also:

ONE_LINE_NEVER, ONE_LINE_IF_EMPTY, ONE_LINE_IF_SINGLE_ITEM, ONE_LINE_ALWAYS, ONE_LINE_PRESERVE, Constant Field Values

FORMATTER_KEEP_SWITCH_CASE_WITH_ARROW_ON_ONE_LINE

```
public static final String  FORMATTER_KEEP_SWITCH_CASE_WITH_ARROW_ON_ONE_LINE
```

```
 FORMATTER / Option to control when a block following a switch case with arrow should be kept on one line
     - option id:         "org.eclipse.jdt.core.formatter.keep_switch_case_with_arrow_on_one_line"
     - possible values:   { ONE_LINE_NEVER, ONE_LINE_IF_EMPTY, ONE_LINE_IF_SINGLE_ITEM,
                            ONE_LINE_ALWAYS, ONE_LINE_PRESERVE }
     - default:           ONE_LINE_NEVER
```

Since:

3.29

See Also:

ONE_LINE_NEVER, ONE_LINE_IF_EMPTY, ONE_LINE_IF_SINGLE_ITEM, ONE_LINE_ALWAYS, ONE_LINE_PRESERVE, Constant Field Values

FORMATTER_KEEP_CODE_BLOCK_ON_ONE_LINE

```
public static final String  FORMATTER_KEEP_CODE_BLOCK_ON_ONE_LINE
```

```
 FORMATTER / Option to control when a code block other than if-then and loop body should
             be kept on one line
     - option id:         "org.eclipse.jdt.core.formatter.keep_code_block_on_one_line"
     - possible values:   { ONE_LINE_NEVER, ONE_LINE_IF_EMPTY }
     - default:           ONE_LINE_NEVER
```

Since:

3.16

See Also:

ONE_LINE_NEVER, ONE_LINE_IF_EMPTY, Constant Field Values

FORMATTER_KEEP_METHOD_BODY_ON_ONE_LINE

```
public static final String  FORMATTER_KEEP_METHOD_BODY_ON_ONE_LINE
```

```
 FORMATTER / Option to control when a method body should be kept on one line
     - option id:         "org.eclipse.jdt.core.formatter.keep_method_body_on_one_line"
     - possible values:   { ONE_LINE_NEVER, ONE_LINE_IF_EMPTY, ONE_LINE_IF_SINGLE_ITEM,
                            ONE_LINE_ALWAYS, ONE_LINE_PRESERVE }
     - default:           ONE_LINE_NEVER
```

Since:

3.16

See Also:

ONE_LINE_NEVER, ONE_LINE_IF_EMPTY, ONE_LINE_IF_SINGLE_ITEM, ONE_LINE_ALWAYS, ONE_LINE_PRESERVE, Constant Field Values

FORMATTER_KEEP_LAMBDA_BODY_BLOCK_ON_ONE_LINE

```
public static final String  FORMATTER_KEEP_LAMBDA_BODY_BLOCK_ON_ONE_LINE
```

```
 FORMATTER / Option to control when a lambda body should be kept on one line
     - option id:         "org.eclipse.jdt.core.formatter.keep_lambda_body_block_on_one_line"
     - possible values:   { ONE_LINE_NEVER, ONE_LINE_IF_EMPTY, ONE_LINE_IF_SINGLE_ITEM,
                            ONE_LINE_ALWAYS, ONE_LINE_PRESERVE }
     - default:           ONE_LINE_NEVER
```

Since:

3.16

See Also:

ONE_LINE_NEVER, ONE_LINE_IF_EMPTY, ONE_LINE_IF_SINGLE_ITEM, ONE_LINE_ALWAYS, ONE_LINE_PRESERVE, Constant Field Values

FORMATTER_KEEP_SIMPLE_GETTER_SETTER_ON_ONE_LINE

```
public static final String  FORMATTER_KEEP_SIMPLE_GETTER_SETTER_ON_ONE_LINE
```

```
 FORMATTER / Option to always keep simple getters and setters on one line, in addition to the
             #FORMATTER_KEEP_METHOD_BODY_ON_ONE_LINE option
     - option id:         "org.eclipse.jdt.core.formatter.keep_simple_getter_setter_on_one_line"
     - possible values:   { TRUE, FALSE }
     - default:           FALSE
```

Since:

3.16

See Also:

TRUE,
FALSE,
FORMATTER_KEEP_METHOD_BODY_ON_ONE_LINE,
Constant Field Values

FORMATTER_KEEP_TYPE_DECLARATION_ON_ONE_LINE

```
public static final String  FORMATTER_KEEP_TYPE_DECLARATION_ON_ONE_LINE
```

```
 FORMATTER / Option to control when a type declaration should be kept on one line
     - option id:         "org.eclipse.jdt.core.formatter.keep_type_declaration_on_one_line"
     - possible values:   { ONE_LINE_NEVER, ONE_LINE_IF_EMPTY, ONE_LINE_IF_SINGLE_ITEM,
                            ONE_LINE_ALWAYS, ONE_LINE_PRESERVE }
     - default:           ONE_LINE_NEVER
```

Since:

3.16, 3.0

See Also:

ONE_LINE_NEVER, ONE_LINE_IF_EMPTY, ONE_LINE_IF_SINGLE_ITEM, ONE_LINE_ALWAYS, ONE_LINE_PRESERVE, Constant Field Values

FORMATTER_KEEP_ANONYMOUS_TYPE_DECLARATION_ON_ONE_LINE

```
public static final String  FORMATTER_KEEP_ANONYMOUS_TYPE_DECLARATION_ON_ONE_LINE
```

```
 FORMATTER / Option to control when an anonymous type declaration should be kept on one line
     - option id:         "org.eclipse.jdt.core.formatter.keep_anonymous_type_declaration_on_one_line"
     - possible values:   { ONE_LINE_NEVER, ONE_LINE_IF_EMPTY, ONE_LINE_IF_SINGLE_ITEM,
                            ONE_LINE_ALWAYS, ONE_LINE_PRESERVE }
     - default:           ONE_LINE_NEVER
```

Since:

3.16

See Also:

ONE_LINE_NEVER, ONE_LINE_IF_EMPTY, ONE_LINE_IF_SINGLE_ITEM, ONE_LINE_ALWAYS, ONE_LINE_PRESERVE, Constant Field Values

FORMATTER_KEEP_ENUM_CONSTANT_DECLARATION_ON_ONE_LINE

```
public static final String  FORMATTER_KEEP_ENUM_CONSTANT_DECLARATION_ON_ONE_LINE
```

```
 FORMATTER / Option to control when an enum constant declaration body should be kept on one line
     - option id:         "org.eclipse.jdt.core.formatter.keep_enum_constant_declaration_on_one_line"
     - possible values:   { ONE_LINE_NEVER, ONE_LINE_IF_EMPTY, ONE_LINE_IF_SINGLE_ITEM,
                            ONE_LINE_ALWAYS, ONE_LINE_PRESERVE }
     - default:           ONE_LINE_NEVER
```

Since:

3.16

See Also:

ONE_LINE_NEVER, ONE_LINE_IF_EMPTY, ONE_LINE_IF_SINGLE_ITEM, ONE_LINE_ALWAYS, ONE_LINE_PRESERVE, Constant Field Values

FORMATTER_KEEP_ENUM_DECLARATION_ON_ONE_LINE

```
public static final String  FORMATTER_KEEP_ENUM_DECLARATION_ON_ONE_LINE
```

```
 FORMATTER / Option to control when an enum declaration should be kept on one line
     - option id:         "org.eclipse.jdt.core.formatter.keep_enum_declaration_on_one_line"
     - possible values:   { ONE_LINE_NEVER, ONE_LINE_IF_EMPTY, ONE_LINE_IF_SINGLE_ITEM,
                            ONE_LINE_ALWAYS, ONE_LINE_PRESERVE }
     - default:           ONE_LINE_NEVER
```

Since:

3.16

See Also:

ONE_LINE_NEVER, ONE_LINE_IF_EMPTY, ONE_LINE_IF_SINGLE_ITEM, ONE_LINE_ALWAYS, ONE_LINE_PRESERVE, Constant Field Values

FORMATTER_KEEP_ANNOTATION_DECLARATION_ON_ONE_LINE

```
public static final String  FORMATTER_KEEP_ANNOTATION_DECLARATION_ON_ONE_LINE
```

```
 FORMATTER / Option to control when an annotation declaration should be kept on one line
     - option id:         "org.eclipse.jdt.core.formatter.keep_annotation_declaration_on_one_line"
     - possible values:   { ONE_LINE_NEVER, ONE_LINE_IF_EMPTY, ONE_LINE_IF_SINGLE_ITEM,
                            ONE_LINE_ALWAYS, ONE_LINE_PRESERVE }
     - default:           ONE_LINE_NEVER
```

Since:

3.16

See Also:

ONE_LINE_NEVER, ONE_LINE_IF_EMPTY, ONE_LINE_IF_SINGLE_ITEM, ONE_LINE_ALWAYS, ONE_LINE_PRESERVE, Constant Field Values

FORMATTER_KEEP_RECORD_DECLARATION_ON_ONE_LINE

```
public static final String  FORMATTER_KEEP_RECORD_DECLARATION_ON_ONE_LINE
```

```
 FORMATTER / Option to control when a record declaration should be kept on one line
     - option id:         "org.eclipse.jdt.core.formatter.keep_record_declaration_on_one_line"
     - possible values:   { ONE_LINE_NEVER, ONE_LINE_IF_EMPTY, ONE_LINE_IF_SINGLE_ITEM,
                            ONE_LINE_ALWAYS, ONE_LINE_PRESERVE }
     - default:           ONE_LINE_NEVER
```

Since:

3.22

See Also:

ONE_LINE_NEVER, ONE_LINE_IF_EMPTY, ONE_LINE_IF_SINGLE_ITEM, ONE_LINE_ALWAYS, ONE_LINE_PRESERVE, Constant Field Values

FORMATTER_KEEP_RECORD_CONSTRUCTOR_ON_ONE_LINE

```
public static final String  FORMATTER_KEEP_RECORD_CONSTRUCTOR_ON_ONE_LINE
```

```
 FORMATTER / Option to control when a record constructor should be kept on one line
     - option id:         "org.eclipse.jdt.core.formatter.keep_record_constructor_on_one_line"
     - possible values:   { ONE_LINE_NEVER, ONE_LINE_IF_EMPTY, ONE_LINE_IF_SINGLE_ITEM,
                            ONE_LINE_ALWAYS, ONE_LINE_PRESERVE }
     - default:           ONE_LINE_NEVER
```

Since:

3.22

See Also:

ONE_LINE_NEVER, ONE_LINE_IF_EMPTY, ONE_LINE_IF_SINGLE_ITEM, ONE_LINE_ALWAYS, ONE_LINE_PRESERVE, Constant Field Values

FORMATTER_LINE_SPLIT

```
public static final String  FORMATTER_LINE_SPLIT
```

```
 FORMATTER / Option to specify the length of the page. Beyond this length, the formatter will try to split the code
     - option id:         "org.eclipse.jdt.core.formatter.lineSplit"
     - possible values:   "<n>", where n is zero or a positive integer
     - default:           "120"
```

Since:

3.0

See Also:

Constant Field Values

FORMATTER_NEVER_INDENT_BLOCK_COMMENTS_ON_FIRST_COLUMN

```
public static final String  FORMATTER_NEVER_INDENT_BLOCK_COMMENTS_ON_FIRST_COLUMN
```

```
 FORMATTER / Option to indent block comments that start on the first column
     - option id:         "org.eclipse.jdt.core.formatter.formatter.never_indent_block_comments_on_first_column"
     - possible values:   { TRUE, FALSE }
     - default:           FALSE
```

Note that this option is ignored if the formatter is created with the mode `ToolFactory.M_FORMAT_NEW`.

Since:

3.3

See Also:

`TRUE`,
`FALSE`,
`ToolFactory.createCodeFormatter(Map, int)`,
Constant Field Values

FORMATTER_NEVER_INDENT_LINE_COMMENTS_ON_FIRST_COLUMN

```
public static final String  FORMATTER_NEVER_INDENT_LINE_COMMENTS_ON_FIRST_COLUMN
```

```
 FORMATTER / Option to indent line comments that start on the first column
     - option id:         "org.eclipse.jdt.core.formatter.formatter.never_indent_line_comments_on_first_column"
     - possible values:   { TRUE, FALSE }
     - default:           FALSE
```

Note that:

- this option is ignored if the formatter is created with the mode `ToolFactory.M_FORMAT_NEW`
- even with this option activated, the formatter still can ignore line comments starting at first column if the option `FORMATTER_COMMENT_FORMAT_LINE_COMMENT_STARTING_ON_FIRST_COLUMN` is set to "false"

Since:

3.3

See Also:

`TRUE`,
`FALSE`,
`ToolFactory.createCodeFormatter(Map, int)`,
Constant Field Values

FORMATTER_NUMBER_OF_EMPTY_LINES_TO_PRESERVE

```
public static final String  FORMATTER_NUMBER_OF_EMPTY_LINES_TO_PRESERVE
```

```
 FORMATTER / Option to specify the number of empty lines to preserve
     - option id:         "org.eclipse.jdt.core.formatter.number_of_empty_lines_to_preserve"
     - possible values:   "<n>", where n is zero or a positive integer
     - default:           "0"
```

Since:

3.0

See Also:

Constant Field Values

FORMATTER_JOIN_WRAPPED_LINES

```
public static final String  FORMATTER_JOIN_WRAPPED_LINES
```

```
 FORMATTER / Option to specify whether the formatter can join wrapped lines or not

                For example, the wrapped lines of method foo return statement in following test case:
                        class X {
                        String foo() {
                        return "select x "
                               + "from y "
                               + "where z=a";
                        }
                        }

                will be preserved by the formatter when the new preference is used
                even if the maximum line width would give it enough space to join the lines.
                Hence produces the following output:
                        class X {
                            String foo() {
```

```
                                return "select x "
                                        + "from y "
                                        + "where z=a";
                                }
                        }
```

```
     - option id:         "org.eclipse.jdt.core.formatter.join_wrapped_lines"
     - possible values:   { TRUE, FALSE }
     - default:           TRUE
```

Since:

3.5

See Also:

Constant Field Values

FORMATTER_JOIN_LINES_IN_COMMENTS

```
public static final String  FORMATTER_JOIN_LINES_IN_COMMENTS
```

```
 FORMATTER / Option to specify whether the formatter can join text lines in comments or not

                For example, the following comment:
                        /**
                         * The foo method.
                         * foo is a substitute for bar.
                         */
                        public class X {
                        }

                will be unchanged by the formatter when this new preference is used,
                even if the maximum line width would give it enough space to join the lines.

     - option id:         "org.eclipse.jdt.core.formatter.join_lines_in_comments"
     - possible values:   { TRUE, FALSE }
     - default:           TRUE
```

Since:

3.5

See Also:

Constant Field Values

FORMATTER_JOIN_LINE_COMMENTS

```
public static final String  FORMATTER_JOIN_LINE_COMMENTS
```

```
 FORMATTER / Option to specify whether the formatter can join consecutive line comments
     - option id:         "org.eclipse.jdt.core.formatter.join_line_comments"
     - possible values:   { TRUE, FALSE }
     - default:           FALSE
```

Since:

3.37

See Also:

Constant Field Values

FORMATTER_PUT_EMPTY_STATEMENT_ON_NEW_LINE

```
public static final String  FORMATTER_PUT_EMPTY_STATEMENT_ON_NEW_LINE
```

```
 FORMATTER / Option to specify whether or not empty statement should be on a new line
     - option id:         "org.eclipse.jdt.core.formatter.put_empty_statement_on_new_line"
     - possible values:   { TRUE, FALSE }
     - default:           FALSE
```

Since:

3.0

See Also:

TRUE, FALSE, Constant Field Values

FORMATTER_TAB_CHAR

```
public static final String  FORMATTER_TAB_CHAR
```

```
 FORMATTER / Option to specify the tabulation size
     - option id:         "org.eclipse.jdt.core.formatter.tabulation.char"
     - possible values:   { TAB, SPACE, MIXED }
     - default:           TAB
```

More values may be added in the future.

Since:

3.0

See Also:

`JavaCore.TAB`, `JavaCore.SPACE`, `MIXED`, Constant Field Values

FORMATTER_TAB_SIZE

```
public static final String  FORMATTER_TAB_SIZE
```

```
 FORMATTER / Option to specify the equivalent number of spaces that represents one tabulation
     - option id:         "org.eclipse.jdt.core.formatter.tabulation.size"
     - possible values:   "<n>", where n is zero or a positive integer
     - default:           "4"
```

Since:

3.0

See Also:

Constant Field Values

FORMATTER_USE_TABS_ONLY_FOR_LEADING_INDENTATIONS

```
public static final String  FORMATTER_USE_TABS_ONLY_FOR_LEADING_INDENTATIONS
```

```
 FORMATTER / Option to use tabulations for indentation and spaces for line wrapping
     - option id:         "org.eclipse.jdt.core.formatter.use_tabs_only_for_leading_indentations"
     - possible values:   { TRUE, FALSE }
     - default:           FALSE
```

Since:

3.1

See Also:

`TRUE`, `FALSE`, Constant Field Values

FORMATTER_WRAP_BEFORE_MULTIPLICATIVE_OPERATOR

```
public static final String  FORMATTER_WRAP_BEFORE_MULTIPLICATIVE_OPERATOR
```

```
 FORMATTER / Option to wrap before the multiplicative operator (*, /, %)
     - option id:         "org.eclipse.jdt.core.formatter.wrap_before_multiplicative_operator"
     - possible values:   { TRUE, FALSE }
     - default:           TRUE
```

This option is used only if the option `FORMATTER_ALIGNMENT_FOR_MULTIPLE_FIELDS` is set.

Since:

3.17

See Also:

`TRUE`, `FALSE`, Constant Field Values

FORMATTER_WRAP_BEFORE_ADDITIVE_OPERATOR

```
public static final String  FORMATTER_WRAP_BEFORE_ADDITIVE_OPERATOR
```

```
 FORMATTER / Option to wrap before the additive operator (+, -)
     - option id:         "org.eclipse.jdt.core.formatter.wrap_before_additive_operator"
     - possible values:   { TRUE, FALSE }
     - default:           TRUE
```

This option is used only if the option `FORMATTER_ALIGNMENT_FOR_ADDITIVE_OPERATOR` is set.

Since:

3.17

See Also:

`TRUE`, `FALSE`, Constant Field Values

FORMATTER_WRAP_BEFORE_ASSERTION_MESSAGE_OPERATOR

```
public static final String  FORMATTER_WRAP_BEFORE_ASSERTION_MESSAGE_OPERATOR
```

```
 FORMATTER / Option to wrap before the assertion message operator
     - option id:         "org.eclipse.jdt.core.formatter.wrap_before_assertion_message_operator"
     - possible values:   { TRUE, FALSE }
     - default:           FALSE
```

Since:

3.23

See Also:

TRUE, FALSE, Constant Field Values

FORMATTER_WRAP_BEFORE_STRING_CONCATENATION

```
public static final String  FORMATTER_WRAP_BEFORE_STRING_CONCATENATION
```

```
 FORMATTER / Option to wrap before the string concatenation operator
     - option id:         "org.eclipse.jdt.core.formatter.wrap_before_string_concatenation"
     - possible values:   { TRUE, FALSE }
     - default:           TRUE
```

This option is used only if the option FORMATTER_ALIGNMENT_FOR_STRING_CONCATENATION is set.

Since:

3.17

See Also:

TRUE, FALSE, Constant Field Values

FORMATTER_WRAP_BEFORE_SHIFT_OPERATOR

```
public static final String  FORMATTER_WRAP_BEFORE_SHIFT_OPERATOR
```

```
 FORMATTER / Option to wrap before the shift operator (<<, >>, >>>)
     - option id:         "org.eclipse.jdt.core.formatter.wrap_before_shift_operator"
     - possible values:   { TRUE, FALSE }
     - default:           TRUE
```

This option is used only if the option FORMATTER_ALIGNMENT_FOR_SHIFT_OPERATOR is set.

Since:

3.17

See Also:

TRUE, FALSE, Constant Field Values

FORMATTER_WRAP_BEFORE_RELATIONAL_OPERATOR

```
public static final String  FORMATTER_WRAP_BEFORE_RELATIONAL_OPERATOR
```

```
 FORMATTER / Option to wrap before the relational operator (<, >, <=, >=, ==, !=)
     - option id:         "org.eclipse.jdt.core.formatter.wrap_before_"
     - possible values:   { TRUE, FALSE }
     - default:           TRUE
```

This option is used only if the option FORMATTER_ALIGNMENT_FOR_RELATIONAL_OPERATOR is set.

Since:

3.17

See Also:

TRUE, FALSE, Constant Field Values

FORMATTER_WRAP_BEFORE_BITWISE_OPERATOR

```
public static final String  FORMATTER_WRAP_BEFORE_BITWISE_OPERATOR
```

```
 FORMATTER / Option to wrap before the bitwise operator (&, ^, |)
     - option id:         "org.eclipse.jdt.core.formatter.wrap_before_bitwise_operator"
     - possible values:   { TRUE, FALSE }
     - default:           TRUE
```

This option is used only if the option FORMATTER_ALIGNMENT_FOR_BITWISE_OPERATOR is set.

Since:

3.17

See Also:

TRUE, FALSE, Constant Field Values

FORMATTER_WRAP_BEFORE_LOGICAL_OPERATOR

```
public static final String  FORMATTER_WRAP_BEFORE_LOGICAL_OPERATOR
```

```
 FORMATTER / Option to wrap before the logical operator (&&, ||)
     - option id:         "org.eclipse.jdt.core.formatter.wrap_before_logical_operator"
     - possible values:   { TRUE, FALSE }
     - default:           TRUE
```

This option is used only if the option FORMATTER_ALIGNMENT_FOR_LOGICAL_OPERATOR is set.

Since:

3.17

See Also:

TRUE, FALSE, Constant Field Values

FORMATTER_WRAP_BEFORE_BINARY_OPERATOR

```
public static final String  FORMATTER_WRAP_BEFORE_BINARY_OPERATOR
```

Deprecated.
Use the new options instead: FORMATTER_WRAP_BEFORE_MULTIPLICATIVE_OPERATOR, FORMATTER_WRAP_BEFORE_ADDITIVE_OPERATOR, FORMATTER_WRAP_BEFORE_STRING_CONCATENATION, FORMATTER_WRAP_BEFORE_BITWISE_OPERATOR, FORMATTER_WRAP_BEFORE_LOGICAL_OPERATOR

```
 FORMATTER / Option to wrap before the binary operator
     - option id:         "org.eclipse.jdt.core.formatter.wrap_before_binary_operator"
     - possible values:   { TRUE, FALSE }
     - default:           TRUE
```

This option is used only if the option FORMATTER_ALIGNMENT_FOR_BINARY_EXPRESSION is set.

Since:

3.3

See Also:

TRUE, FALSE, Constant Field Values

FORMATTER_WRAP_BEFORE_OR_OPERATOR_MULTICATCH

```
public static final String  FORMATTER_WRAP_BEFORE_OR_OPERATOR_MULTICATCH
```

```
 FORMATTER / Option to wrap before the '|' operator in multi-catch statements
     - option id:         "org.eclipse.jdt.core.formatter.wrap_before_or_operator_multicatch"
     - possible values:   { TRUE, FALSE }
     - default:           TRUE
```

This option is used only if the option FORMATTER_ALIGNMENT_FOR_UNION_TYPE_IN_MULTICATCH is set.

Since:

3.7.1

See Also:

TRUE, FALSE, Constant Field Values

FORMATTER_WRAP_BEFORE_CONDITIONAL_OPERATOR

```
public static final String  FORMATTER_WRAP_BEFORE_CONDITIONAL_OPERATOR
```

```
 FORMATTER / Option to wrap before the '?' and ':' operators in conditional expressions
     - option id:         "org.eclipse.jdt.core.formatter.wrap_before_conditional_operator"
     - possible values:   { TRUE, FALSE }
     - default:           TRUE
```

This option is used only if the option FORMATTER_ALIGNMENT_FOR_CONDITIONAL_EXPRESSION is set.

Since:

3.12

See Also:

TRUE, FALSE, Constant Field Values

FORMATTER_WRAP_BEFORE_SWITCH_CASE_ARROW_OPERATOR

```
public static final String  FORMATTER_WRAP_BEFORE_SWITCH_CASE_ARROW_OPERATOR
```

```
 FORMATTER / Option to wrap before the arrow operator (->) in switch case
     - option id:         "org.eclipse.jdt.core.formatter.wrap_before_switch_case_arrow_operator"
     - possible values:   { TRUE, FALSE }
     - default:           FALSE
```

This option is used only if the option FORMATTER_ALIGNMENT_FOR_SWITCH_CASE_WITH_ARROW is set.

Since:

3.29

See Also:

TRUE, FALSE, Constant Field Values

FORMATTER_WRAP_BEFORE_ASSIGNMENT_OPERATOR

```
public static final String  FORMATTER_WRAP_BEFORE_ASSIGNMENT_OPERATOR
```

```
 FORMATTER / Option to wrap before the assignment operator
     - option id:         "org.eclipse.jdt.core.formatter.wrap_before_assignment_operator"
     - possible values:   { TRUE, FALSE }
     - default:           TRUE
```

This option is used only if the option FORMATTER_ALIGNMENT_FOR_ASSIGNMENT is set.

Since:

3.12

See Also:

TRUE, FALSE, Constant Field Values

FORMATTER_WRAP_OUTER_EXPRESSIONS_WHEN_NESTED

```
public static final String  FORMATTER_WRAP_OUTER_EXPRESSIONS_WHEN_NESTED
```

```
 FORMATTER / Option to wrap outer expressions in nested expressions
     - option id:         "org.eclipse.jdt.core.formatter.wrap_outer_expressions_when_nested"
     - possible values:   { TRUE, FALSE }
     - default:           TRUE
```

This option changes the formatter behavior when nested method calls are encountered. Since 3.6, the formatter tries to wrap outermost method calls first to have a better output.

For example, let's say we are using the Eclipse built-in profile with a max line width=40+space for tab policy. Then consider the following snippet:

```
 public class X01 {
     void test() {
         foo(bar(1, 2, 3, 4), bar(5, 6, 7, 8));
     }
 }
```

With this new strategy, the formatter will wrap the line earlier, between the arguments of the message call for this example, and then it will allow to keep each nested call on a single line.

Hence, the output will be:

```
 public class X01 {
     void test() {
         foo(bar(1, 2, 3, 4),
```

```
            bar(5, 6, 7, 8));
    }
}
```

Important notes:

1. This new behavior is automatically activated (i.e. the default value for this preference is `TRUE`). If the backward compatibility regarding previous versions' formatter behavior (i.e. before 3.6 version) is necessary, then the preference needs to be set to `FALSE` to retrieve the previous formatter behavior.
2. The new strategy currently only applies to nested method calls, but might be extended to other nested expressions in future versions

Since:

3.6

See Also:

`TRUE`, `FALSE`, Constant Field Values

INDENT_BY_ONE

```
public static final int INDENT_BY_ONE
```

FORMATTER / The wrapping is done by indenting by one compare to the current indentation.

Since:

3.0

See Also:

Constant Field Values

INDENT_DEFAULT

```
public static final int INDENT_DEFAULT
```

FORMATTER / The wrapping is done by using the current indentation.

Since:

3.0

See Also:

Constant Field Values

INDENT_ON_COLUMN

```
public static final int INDENT_ON_COLUMN
```

FORMATTER / The wrapping is done by indenting on column under the splitting location.

Since:

3.0

See Also:

Constant Field Values

INDENT_PRESERVE

```
public static final int INDENT_PRESERVE
```

FORMATTER / Indentation is not touched, it's preserved from original source.

Since:

3.20

See Also:

Constant Field Values

MIXED

```
public static final String  MIXED
```

```
 FORMATTER / Possible value for the option FORMATTER_TAB_CHAR
```

Since:

3.1

See Also:

JavaCore.TAB, JavaCore.SPACE, FORMATTER_TAB_CHAR, Constant Field Values

NEXT_LINE

```
public static final String  NEXT_LINE
```

```
 FORMATTER / Value to set a brace location at the start of the next line with
             the right indentation.
```

Since:

3.0

See Also:

FORMATTER_BRACE_POSITION_FOR_ANONYMOUS_TYPE_DECLARATION,
FORMATTER_BRACE_POSITION_FOR_ARRAY_INITIALIZER,
FORMATTER_BRACE_POSITION_FOR_BLOCK,
FORMATTER_BRACE_POSITION_FOR_CONSTRUCTOR_DECLARATION,
FORMATTER_BRACE_POSITION_FOR_METHOD_DECLARATION,
FORMATTER_BRACE_POSITION_FOR_RECORD_CONSTRUCTOR,
FORMATTER_BRACE_POSITION_FOR_RECORD_DECLARATION,
FORMATTER_BRACE_POSITION_FOR_SWITCH,
FORMATTER_BRACE_POSITION_FOR_TYPE_DECLARATION,
FORMATTER_BRACE_POSITION_FOR_LAMBDA_BODY,
Constant Field Values

NEXT_LINE_ON_WRAP

```
public static final String  NEXT_LINE_ON_WRAP
```

```
 FORMATTER / Value to set a brace location at the start of the next line if a wrapping
             occured.
```

Since:

3.0

See Also:

FORMATTER_BRACE_POSITION_FOR_ANONYMOUS_TYPE_DECLARATION,
FORMATTER_BRACE_POSITION_FOR_ARRAY_INITIALIZER,
FORMATTER_BRACE_POSITION_FOR_BLOCK,
FORMATTER_BRACE_POSITION_FOR_CONSTRUCTOR_DECLARATION,
FORMATTER_BRACE_POSITION_FOR_METHOD_DECLARATION,
FORMATTER_BRACE_POSITION_FOR_RECORD_CONSTRUCTOR,
FORMATTER_BRACE_POSITION_FOR_RECORD_DECLARATION,
FORMATTER_BRACE_POSITION_FOR_SWITCH,
FORMATTER_BRACE_POSITION_FOR_TYPE_DECLARATION,
FORMATTER_BRACE_POSITION_FOR_LAMBDA_BODY,
Constant Field Values

NEXT_LINE_SHIFTED

```
public static final String  NEXT_LINE_SHIFTED
```

```
 FORMATTER / Value to set a brace location at the start of the next line with
             an extra indentation.
```

Since:

3.0

See Also:

FORMATTER_BRACE_POSITION_FOR_ANONYMOUS_TYPE_DECLARATION,
FORMATTER_BRACE_POSITION_FOR_ARRAY_INITIALIZER,
FORMATTER_BRACE_POSITION_FOR_BLOCK,
FORMATTER_BRACE_POSITION_FOR_CONSTRUCTOR_DECLARATION,
FORMATTER_BRACE_POSITION_FOR_METHOD_DECLARATION,
FORMATTER_BRACE_POSITION_FOR_RECORD_CONSTRUCTOR,
FORMATTER_BRACE_POSITION_FOR_RECORD_DECLARATION,
FORMATTER_BRACE_POSITION_FOR_SWITCH,

FORMATTER_BRACE_POSITION_FOR_TYPE_DECLARATION,
FORMATTER_BRACE_POSITION_FOR_LAMBDA_BODY,
Constant Field Values

COMMON_LINES

```
public static final String  COMMON_LINES
```

```
 FORMATTER / Value to set opening and closing parentheses location in common lines with
             their contents (or simply a single line if the parentheses are empty).
```

Since:

3.12

See Also:

FORMATTER_PARENTHESES_POSITIONS_IN_METHOD_DECLARATION,
FORMATTER_PARENTHESES_POSITIONS_IN_METHOD_INVOCATION,
FORMATTER_PARENTHESES_POSITIONS_IN_ENUM_CONSTANT_DECLARATION,
FORMATTER_PARENTHESES_POSITIONS_IN_RECORD_DECLARATION,
FORMATTER_PARENTHESES_POSITIONS_IN_IF_WHILE_STATEMENT,
FORMATTER_PARENTHESES_POSITIONS_IN_FOR_STATEMENT,
FORMATTER_PARENTHESES_POSITIONS_IN_SWITCH_STATEMENT,
FORMATTER_PARENTHESES_POSITIONS_IN_TRY_CLAUSE,
FORMATTER_PARENTHESES_POSITIONS_IN_CATCH_CLAUSE,
FORMATTER_PARENTHESES_POSITIONS_IN_ANNOTATION,
FORMATTER_PARENTHESES_POSITIONS_IN_LAMBDA_DECLARATION,
Constant Field Values

SEPARATE_LINES_IF_NOT_EMPTY

```
public static final String  SEPARATE_LINES_IF_NOT_EMPTY
```

```
 FORMATTER / Value to set opening and closing parentheses location on a common line
             if the parentheses are empty and otherwise in separate lines from their contents.
```

Since:

3.12

See Also:

FORMATTER_PARENTHESES_POSITIONS_IN_METHOD_DECLARATION,
FORMATTER_PARENTHESES_POSITIONS_IN_METHOD_INVOCATION,
FORMATTER_PARENTHESES_POSITIONS_IN_ENUM_CONSTANT_DECLARATION,
FORMATTER_PARENTHESES_POSITIONS_IN_RECORD_DECLARATION,
FORMATTER_PARENTHESES_POSITIONS_IN_ANNOTATION,
FORMATTER_PARENTHESES_POSITIONS_IN_LAMBDA_DECLARATION,
Constant Field Values

SEPARATE_LINES_IF_WRAPPED

```
public static final String  SEPARATE_LINES_IF_WRAPPED
```

```
 FORMATTER / Value to set opening and closing parentheses location on separate lines from their
             contents if the contents are wrapped, and in common line if they fit in line width.
```

Since:

3.12

See Also:

FORMATTER_PARENTHESES_POSITIONS_IN_METHOD_DECLARATION,
FORMATTER_PARENTHESES_POSITIONS_IN_METHOD_INVOCATION,
FORMATTER_PARENTHESES_POSITIONS_IN_ENUM_CONSTANT_DECLARATION,
FORMATTER_PARENTHESES_POSITIONS_IN_RECORD_DECLARATION,
FORMATTER_PARENTHESES_POSITIONS_IN_IF_WHILE_STATEMENT,
FORMATTER_PARENTHESES_POSITIONS_IN_FOR_STATEMENT,
FORMATTER_PARENTHESES_POSITIONS_IN_SWITCH_STATEMENT,
FORMATTER_PARENTHESES_POSITIONS_IN_TRY_CLAUSE,
FORMATTER_PARENTHESES_POSITIONS_IN_CATCH_CLAUSE,
FORMATTER_PARENTHESES_POSITIONS_IN_ANNOTATION,
FORMATTER_PARENTHESES_POSITIONS_IN_LAMBDA_DECLARATION,
Constant Field Values

SEPARATE_LINES

```
public static final String  SEPARATE_LINES
```

```
 FORMATTER / Value to set parentheses location on separate lines from their contents,
             that is put a line break after the opening parenthesis and before
```

the closing parenthesis.

Since:

3.12

See Also:

FORMATTER_PARENTHESES_POSITIONS_IN_METHOD_DECLARATION,
FORMATTER_PARENTHESES_POSITIONS_IN_METHOD_INVOCATION,
FORMATTER_PARENTHESES_POSITIONS_IN_ENUM_CONSTANT_DECLARATION,
FORMATTER_PARENTHESES_POSITIONS_IN_RECORD_DECLARATION,
FORMATTER_PARENTHESES_POSITIONS_IN_IF_WHILE_STATEMENT,
FORMATTER_PARENTHESES_POSITIONS_IN_FOR_STATEMENT,
FORMATTER_PARENTHESES_POSITIONS_IN_SWITCH_STATEMENT,
FORMATTER_PARENTHESES_POSITIONS_IN_TRY_CLAUSE,
FORMATTER_PARENTHESES_POSITIONS_IN_CATCH_CLAUSE,
FORMATTER_PARENTHESES_POSITIONS_IN_ANNOTATION,
FORMATTER_PARENTHESES_POSITIONS_IN_LAMBDA_DECLARATION,
Constant Field Values

PRESERVE_POSITIONS

```
public static final String  PRESERVE_POSITIONS
```

FORMATTER / Value to set opening and closing parentheses location to be preserved
from the original source.

Since:

3.12

See Also:

FORMATTER_PARENTHESES_POSITIONS_IN_METHOD_DECLARATION,
FORMATTER_PARENTHESES_POSITIONS_IN_METHOD_INVOCATION,
FORMATTER_PARENTHESES_POSITIONS_IN_ENUM_CONSTANT_DECLARATION,
FORMATTER_PARENTHESES_POSITIONS_IN_RECORD_DECLARATION,
FORMATTER_PARENTHESES_POSITIONS_IN_IF_WHILE_STATEMENT,
FORMATTER_PARENTHESES_POSITIONS_IN_FOR_STATEMENT,
FORMATTER_PARENTHESES_POSITIONS_IN_SWITCH_STATEMENT,
FORMATTER_PARENTHESES_POSITIONS_IN_TRY_CLAUSE,
FORMATTER_PARENTHESES_POSITIONS_IN_CATCH_CLAUSE,
FORMATTER_PARENTHESES_POSITIONS_IN_ANNOTATION,
FORMATTER_PARENTHESES_POSITIONS_IN_LAMBDA_DECLARATION,
Constant Field Values

ONE_LINE_NEVER

```
public static final String  ONE_LINE_NEVER
```

FORMATTER / Value to never keep braced code on one line.

Since:

3.16

See Also:

FORMATTER_KEEP_LOOP_BODY_BLOCK_ON_ONE_LINE,
FORMATTER_KEEP_IF_THEN_BODY_BLOCK_ON_ONE_LINE,
FORMATTER_KEEP_CODE_BLOCK_ON_ONE_LINE,
FORMATTER_KEEP_METHOD_BODY_ON_ONE_LINE,
FORMATTER_KEEP_LAMBDA_BODY_BLOCK_ON_ONE_LINE,
FORMATTER_KEEP_TYPE_DECLARATION_ON_ONE_LINE,
FORMATTER_KEEP_ANONYMOUS_TYPE_DECLARATION_ON_ONE_LINE,
FORMATTER_KEEP_ENUM_CONSTANT_DECLARATION_ON_ONE_LINE,
FORMATTER_KEEP_ENUM_DECLARATION_ON_ONE_LINE,
FORMATTER_KEEP_ANNOTATION_DECLARATION_ON_ONE_LINE,
Constant Field Values

ONE_LINE_IF_EMPTY

```
public static final String  ONE_LINE_IF_EMPTY
```

FORMATTER / Value to keep braced code on one line only if it's empty.

Since:

3.16

See Also:

FORMATTER_KEEP_LOOP_BODY_BLOCK_ON_ONE_LINE,
FORMATTER_KEEP_IF_THEN_BODY_BLOCK_ON_ONE_LINE,

FORMATTER_KEEP_CODE_BLOCK_ON_ONE_LINE,
FORMATTER_KEEP_METHOD_BODY_ON_ONE_LINE,
FORMATTER_KEEP_LAMBDA_BODY_BLOCK_ON_ONE_LINE,
FORMATTER_KEEP_TYPE_DECLARATION_ON_ONE_LINE,
FORMATTER_KEEP_ANONYMOUS_TYPE_DECLARATION_ON_ONE_LINE,
FORMATTER_KEEP_ENUM_CONSTANT_DECLARATION_ON_ONE_LINE,
FORMATTER_KEEP_ENUM_DECLARATION_ON_ONE_LINE,
FORMATTER_KEEP_ANNOTATION_DECLARATION_ON_ONE_LINE,
Constant Field Values

ONE_LINE_IF_SINGLE_ITEM

```
public static final String  ONE_LINE_IF_SINGLE_ITEM
```

```
 FORMATTER / Value to keep braced code on one line if it contains at most a single
             item.
```

Since:

3.16

See Also:

FORMATTER_KEEP_LOOP_BODY_BLOCK_ON_ONE_LINE,
FORMATTER_KEEP_IF_THEN_BODY_BLOCK_ON_ONE_LINE,
FORMATTER_KEEP_METHOD_BODY_ON_ONE_LINE,
FORMATTER_KEEP_LAMBDA_BODY_BLOCK_ON_ONE_LINE,
FORMATTER_KEEP_TYPE_DECLARATION_ON_ONE_LINE,
FORMATTER_KEEP_ANONYMOUS_TYPE_DECLARATION_ON_ONE_LINE,
FORMATTER_KEEP_ENUM_CONSTANT_DECLARATION_ON_ONE_LINE,
FORMATTER_KEEP_ENUM_DECLARATION_ON_ONE_LINE,
FORMATTER_KEEP_ANNOTATION_DECLARATION_ON_ONE_LINE,
Constant Field Values

ONE_LINE_ALWAYS

```
public static final String  ONE_LINE_ALWAYS
```

```
 FORMATTER / Value to always keep braced code on one line, as long as it doesn't
             exceed the line width limit.
```

Since:

3.16

See Also:

FORMATTER_KEEP_LOOP_BODY_BLOCK_ON_ONE_LINE,
FORMATTER_KEEP_IF_THEN_BODY_BLOCK_ON_ONE_LINE,
FORMATTER_KEEP_METHOD_BODY_ON_ONE_LINE,
FORMATTER_KEEP_LAMBDA_BODY_BLOCK_ON_ONE_LINE,
FORMATTER_KEEP_TYPE_DECLARATION_ON_ONE_LINE,
FORMATTER_KEEP_ANONYMOUS_TYPE_DECLARATION_ON_ONE_LINE,
FORMATTER_KEEP_ENUM_CONSTANT_DECLARATION_ON_ONE_LINE,
FORMATTER_KEEP_ENUM_DECLARATION_ON_ONE_LINE,
FORMATTER_KEEP_ANNOTATION_DECLARATION_ON_ONE_LINE,
Constant Field Values

ONE_LINE_PRESERVE

```
public static final String  ONE_LINE_PRESERVE
```

```
 FORMATTER / Value to keep braced code on one line as long as it doesn't exceed the
             line width limit and it was already in one line in the original source.
```

Since:

3.16

See Also:

FORMATTER_KEEP_LOOP_BODY_BLOCK_ON_ONE_LINE,
FORMATTER_KEEP_IF_THEN_BODY_BLOCK_ON_ONE_LINE,
FORMATTER_KEEP_METHOD_BODY_ON_ONE_LINE,
FORMATTER_KEEP_LAMBDA_BODY_BLOCK_ON_ONE_LINE,
FORMATTER_KEEP_TYPE_DECLARATION_ON_ONE_LINE,
FORMATTER_KEEP_ANONYMOUS_TYPE_DECLARATION_ON_ONE_LINE,
FORMATTER_KEEP_ENUM_CONSTANT_DECLARATION_ON_ONE_LINE,
FORMATTER_KEEP_ENUM_DECLARATION_ON_ONE_LINE,
FORMATTER_KEEP_ANNOTATION_DECLARATION_ON_ONE_LINE,
Constant Field Values

TRUE

```
public static final String  TRUE
```

FORMATTER / Value to set an option to true.

Since:

3.0

See Also:

Constant Field Values

WRAP_COMPACT

```
public static final int WRAP_COMPACT
```

FORMATTER / The wrapping is done using as few lines as possible.

Since:

3.0

See Also:

Constant Field Values

WRAP_COMPACT_FIRST_BREAK

```
public static final int WRAP_COMPACT_FIRST_BREAK
```

FORMATTER / The wrapping is done putting the first element on a new
line and then wrapping next elements using as few lines as possible.

Since:

3.0

See Also:

Constant Field Values

WRAP_NEXT_PER_LINE

```
public static final int WRAP_NEXT_PER_LINE
```

FORMATTER / The wrapping is done by putting each element on its own line
except the first element.

Since:

3.0

See Also:

Constant Field Values

WRAP_NEXT_SHIFTED

```
public static final int WRAP_NEXT_SHIFTED
```

FORMATTER / The wrapping is done by putting each element on its own line.
All elements are indented by one except the first element.

Since:

3.0

See Also:

Constant Field Values

WRAP_NO_SPLIT

```
public static final int WRAP_NO_SPLIT
```

FORMATTER / Value to disable alignment.

Since:

3.0

See Also:

Constant Field Values

WRAP_ONE_PER_LINE

```
public static final int WRAP_ONE_PER_LINE
```

FORMATTER / The wrapping is done by putting each element on its own line.

Since:

3.0

See Also:

Constant Field Values

Constructor Details

DefaultCodeFormatterConstants

```
public DefaultCodeFormatterConstants()
```

Method Details

createAlignmentValue

```
public static String  createAlignmentValue(boolean forceSplit,
                                           int wrapStyle,
                                           int indentStyle)
```

Create a new alignment value according to the given values. This must be used to set up the alignment options.

Parameters:

`forceSplit` - the given force value

`wrapStyle` - the given wrapping style

`indentStyle` - the given indent style

Returns:

the new alignment value

createAlignmentValue

```
public static String  createAlignmentValue(boolean forceSplit,
                                           int wrapStyle)
```

Create a new alignment value according to the given values. This must be used to set up the alignment options that don't allow for various indent styles.

Parameters:

`forceSplit` - the given force value

`wrapStyle` - the given wrapping style

Returns:

the new alignment value

Since:

3.24

getEclipse21Settings

```
public static Map <String ,String > getEclipse21Settings()
```

Returns the formatter settings that most closely approximate the default formatter settings of Eclipse version 2.1.

Returns:

the Eclipse 2.1 settings

Since:

3.0

getEclipseDefaultSettings

```
public static Map <String ,String > getEclipseDefaultSettings()
```

Returns the default Eclipse formatter settings

Returns:

the Eclipse default settings

Since:

3.1

getForceWrapping

```
public static boolean getForceWrapping(String  value)
```

Return the force value of the given alignment value. The given alignment value should be created using the `createAlignmentValue(boolean, int, int)` API.

Parameters:

`value` - the given alignment value

Returns:

the force value of the given alignment value

Throws:

`IllegalArgumentException` - if the given alignment value is null, or if it doesn't have a valid format.

See Also:

`createAlignmentValue(boolean, int, int)`

getIndentStyle

```
public static int getIndentStyle(String  value)
```

Return the indentation style of the given alignment value. The given alignment value should be created using the `createAlignmentValue(boolean, int, int)` API.

Parameters:

`value` - the given alignment value

Returns:

the indentation style of the given alignment value

Throws:

`IllegalArgumentException` - if the given alignment value is null, or if it doesn't have a valid format.

See Also:

`createAlignmentValue(boolean, int, int)`

getJavaConventionsSettings

```
public static Map <String ,String > getJavaConventionsSettings()
```

Returns the settings according to the Java conventions.

Returns:

the settings according to the Java conventions

Since:

3.0

getWrappingStyle

```
public static int getWrappingStyle(String  value)
```

Return the wrapping style of the given alignment value. The given alignment value should be created using the `createAlignmentValue(boolean, int, int)` API.

Parameters:

`value` - the given alignment value

Returns:

the wrapping style of the given alignment value

Throws:

`IllegalArgumentException` - if the given alignment value is null, or if it doesn't have a valid format.

See Also:

`createAlignmentValue(boolean, int, int)`

setForceWrapping

```
public static String  setForceWrapping(String  value,
                                       boolean force)
```

Set the force value of the given alignment value and return the new value. The given alignment value should be created using the `createAlignmentValue(boolean, int, int)` API.

Parameters:

`value` - the given alignment value

`force` - the given force value

Returns:

the new alignment value

Throws:

`IllegalArgumentException` - if the given alignment value is null, or if it doesn't have a valid format.

See Also:

`createAlignmentValue(boolean, int, int)`

setIndentStyle

```
public static String  setIndentStyle(String  value,
                                     int indentStyle)
```

Set the indentation style of the given alignment value and return the new value. The given value should be created using the `createAlignmentValue(boolean, int, int)` API.

Parameters:

`value` - the given alignment value

`indentStyle` - the given indentation style

Returns:

the new alignment value

Throws:

`IllegalArgumentException` - if the given alignment value is null, if the given indentation style is not one of the possible indentation styles, or if the given alignment value doesn't have a valid format.

See Also:

`INDENT_BY_ONE`,
`INDENT_DEFAULT`,
`INDENT_ON_COLUMN`,
`createAlignmentValue(boolean, int, int)`

setWrappingStyle

```
public static String  setWrappingStyle(String  value,
                                       int wrappingStyle)
```

Set the wrapping style of the given alignment value and return the new value. The given value should be created using the `createAlignmentValue(boolean, int, int)` API.

Parameters:

`value` - the given alignment value

`wrappingStyle` - the given wrapping style

Returns:

the new alignment value

Throws:

`IllegalArgumentException` - if the given alignment value is null, if the given wrapping style is not one of the possible wrapping styles, or if the given alignment value doesn't have a valid format.

See Also:

`WRAP_COMPACT`,
`WRAP_COMPACT_FIRST_BREAK`,
`WRAP_NEXT_PER_LINE`,
`WRAP_NEXT_SHIFTED`,
`WRAP_NO_SPLIT`,
`WRAP_ONE_PER_LINE`,
`createAlignmentValue(boolean, int, int)`