

Convenciones de formato de EditorConfig en C++

Artículo • 09/08/2022 • Tiempo de lectura: 6 minutos

Se aplica a:  Visual Studio Visual Studio  para Mac  Visual Studio Code

El formateador de Visual Studio C++ tiene un amplio conjunto de valores configurables que se pueden aplicar de manera global. Para establecer valores de formato de C++ para un área de trabajo específica, use [clangformat](#) o [EditorConfig](#). Tanto Visual Studio como Visual Studio Code tienen compatibilidad integrada con EditorConfig para cada una de las opciones de formato globales de Visual Studio C++, con la configuración de EditorConfig con prioridad. Esto significa que puede agregar archivos de EditorConfig al área de trabajo para configurar el formato de C++ en un nivel más pormenorizado y aplicar un estilo de código coherente para todos los usuarios que colaboran en el proyecto.

Convenciones de formato de C++

Los valores de EditorConfig de formato de C++ tienen el prefijo `cpp_`. Este es un ejemplo del aspecto del archivo EditorConfig:

ini
<pre>[*. {c++,cc,cpp,cxx,h,h++,hh,hpp,hxx,inl,ipp,tlh,tli}] cpp_indent_case_contents_when_block = true cpp_new_line_before_open_brace_namespace = same_line</pre>

En el resto de este documento se enumeran todos los valores de formato de C++ de EditorConfig compatibles con Visual Studio y VS Code.

Configuración de sangría

Aplicar sangría a las llaves

- Nombre: `cpp_indent_braces`
- Valores: `true`, `false`

Aplicar sangría a cada línea relativa a

- Nombre: `cpp_indent_multi_line_relative_to`
- Valores:

- `outermost_parenthesis`. Cuando se escribe una nueva línea, se aplica la sangría en relación al paréntesis de apertura externo.
- `innermost_parenthesis`. Cuando se escribe una nueva línea, se aplica la sangría en relación al paréntesis de apertura interno.
- `statement_begin`. Cuando se escribe una nueva línea, se aplica la sangría en relación al principio de la instrucción actual.

Entre paréntesis, alinear las líneas nuevas al escribirlas

- Nombre: `cpp_indent_within_parentheses`
- Valores:
 - `align_to_parenthesis`. Alinea el contenido con el paréntesis de apertura.
 - `indent`. Aplica sangría a las líneas nuevas.

En el código existente, no usar la configuración para alinear las líneas nuevas entre paréntesis

- Nombre: `cpp_indent_preserve_within_parentheses`
- Valores: `true`, `false`

Aplicar sangría al contenido Case

- Nombre: `cpp_indent_case_contents`
- Valores: `true`, `false`

Aplicar sangría a etiquetas Case

- Nombre: `cpp_indent_case_labels`
- Valores: `true`, `false`

Aplicar sangría a las llaves siguiendo una instrucción Case

- Nombre: `cpp_indent_case_contents_when_block`
- Valores: `true`, `false`

Aplicar sangría a las llaves de expresiones lambda usadas como parámetros

- Nombre: `cpp_indent_lambda_braces_when_parameter`
- Valores: `true`, `false`

Posición de las etiquetas goto

- Nombre: `cpp_indent_goto_labels`
- Valores:
 - `one_left`. Una sangría a la izquierda
 - `leftmost_column`. Mover a la primera columna de la izquierda
 - `none`. Dejar con sangría aplicada

Posición de las directivas de preprocesador

- Nombre: `cpp_indent_preprocessor`
- Valores:
 - `one_left`. Una sangría a la izquierda
 - `leftmost_column`. Mover a la primera columna de la izquierda
 - `none`. Dejar con sangría aplicada

Aplicar sangría a los especificadores de acceso

- Nombre: `cpp_indent_access_specifiers`
- Valores: `true`, `false`

Aplicar sangría al contenido del espacio de nombres

- Nombre: `cpp_indent_namespace_contents`
- Valores: `true`, `false`

Conservar la sangría de los comentarios

- Nombre: `cpp_indent_preserve_comments`
- Valores: `true`, `false`

Configuración de nueva línea

Posición de llaves de apertura para espacios de nombres

- Nombre: `cpp_new_line_before_open_brace_namespace`
- Valores:
 - `new_line`. Mover a una nueva línea
 - `same_line`. Mantener en la misma línea, pero agregar un espacio delante
 - `ignore`. No reubicar automáticamente

Posición de llaves de apertura para tipos

- Nombre: `cpp_new_line_before_open_brace_type`
- Valores:
 - `new_line`. Mover a una nueva línea
 - `same_line`. Mantener en la misma línea, pero agregar un espacio delante
 - `ignore`. No reubicar automáticamente

Posición de llaves de apertura para funciones

- Nombre: `cpp_new_line_before_open_brace_function`
- Valores:
 - `new_line`. Mover a una nueva línea

- `same_line`. Mantener en la misma línea, pero agregar un espacio delante
- `ignore`. No reubicar automáticamente

Posición de llaves de apertura para bloques de control

- Nombre: `cpp_new_line_before_open_brace_block`
- Valores:
 - `new_line`. Mover a una nueva línea
 - `same_line`. Mantener en la misma línea, pero agregar un espacio delante
 - `ignore`. No reubicar automáticamente

Posición de llaves de apertura para expresiones lambda

- Nombre: `cpp_new_line_before_open_brace_lambda`
- Valores:
 - `new_line`. Mover a una nueva línea
 - `same_line`. Mantener en la misma línea, pero agregar un espacio delante
 - `ignore`. No reubicar automáticamente

Colocar llaves de ámbito en líneas separadas

- Nombre: `cpp_new_line_scope_braces_on_separate_lines`
- Valores: `true`, `false`

Para los tipos vacíos, mover las llaves de cierre a la misma línea que las de apertura

- Nombre: `cpp_new_line_close_brace_same_line_empty_type`
- Valores: `true`, `false`

Para los cuerpos de función vacíos, mover las llaves de cierre a la misma línea que las de apertura

- Nombre: `cpp_new_line_close_brace_same_line_empty_function`
- Valores: `true`, `false`

Colocar "catch" y palabras clave similares en una nueva línea

- Nombre: `cpp_new_line_before_catch`
- Valores: `true`, `false`

Colocar "else" en una nueva línea

- Nombre: `cpp_new_line_before_else`
- Valores: `true`, `false`

Colocar "while" de un bucle do-while en una nueva línea

- Nombre: `cpp_new_line_before_while_in_do_while`

- Valores: true, false

Configuración de espaciado

Espaciado entre nombres de funciones y paréntesis de apertura de listas de argumentos

- Nombre: `cpp_space_before_function_open_parenthesis`
- Valores:
 - `insert`. Insertar un espacio
 - `remove`. Quitar espacios
 - `ignore`. No cambiar los espacios

Insertar espacio entre los paréntesis de una lista de argumentos

- Nombre: `cpp_space_within_parameter_list_parentheses` Valores: true, false

Insertar espacio entre paréntesis cuando la lista de argumentos está vacía

- Nombre: `cpp_space_between_empty_parameter_list_parentheses`
- Valores: true, false

Insertar espacio entre palabra clave y paréntesis de apertura en instrucciones de flujo de control

- Nombre: `cpp_space_after_keywords_in_control_flow_statements`
- Valores: true, false

Insertar espacio entre los paréntesis de una instrucción de control

- Nombre: `cpp_space_within_control_flow_statement_parentheses`
- Valores: true, false

Insertar espacio delante del paréntesis de apertura de las listas de argumentos lambda

- Nombre: `cpp_space_before_lambda_open_parenthesis`
- Valores: true, false

Insertar espacio entre los paréntesis de una conversión de estilo de C

- Nombre: `cpp_space_within_cast_parentheses`
- Valores: true, false

Insertar espacio tras paréntesis de cierre de conversión de estilo de C

- Nombre: `cpp_space_after_cast_close_parenthesis`
- Valores: true, false

Insertar espacio entre los paréntesis de una expresión incluida entre paréntesis

- Nombre: `cpp_space_within_expression_parentheses`
- Valores: `true`, `false`

Insertar espacio delante de las llaves de apertura de los bloques

- Nombre: `cpp_space_before_block_open_brace`
- Valores: `true`, `false`

Insertar espacio entre llaves vacías

- Nombre: `cpp_space_between_empty_braces`
- Valores: `true`, `false`

Insertar espacio delante de la llave de apertura de inicialización uniforme y listas de inicializadores

- Nombre: `cpp_space_before_initializer_list_open_brace`
- Valores: `true`, `false`

Insertar espacio entre llaves de inicialización uniforme y listas de inicializadores

- Nombre: `cpp_space_within_initializer_list_braces`
- Valores: `true`, `false`

Conservar los espacios dentro de las listas uniformes de inicialización y del inicializador

- Nombre: `cpp_space_preserve_in_initializer_list`
- Valores: `true`, `false`

Insertar espacio delante de corchetes de apertura

- Nombre: `cpp_space_before_open_square_bracket`
- Valores: `true`, `false`

Insertar espacio entre corchetes

- Nombre: `cpp_space_within_square_brackets`
- Valores: `true`, `false`

Insertar espacio delante de corchetes vacíos

- Nombre: `cpp_space_before_empty_square_brackets`
- Valores: `true`, `false`

Insertar espacio entre corchetes vacíos

- Nombre: `cpp_space_between_empty_square_brackets`
- Valores: `true`, `false`

Agrupar los corchetes en las matrices multidimensionales

- Nombre: `cpp_space_group_square_brackets`
- Valores: `true`, `false`

Insertar espacio entre corchetes para expresiones lambda

- Nombre: `cpp_space_within_lambda_brackets`
- Valores: `true`, `false`

SpaceBetweenEmptyLambdaBrackets

- Nombre: `cpp_space_between_empty_lambda_brackets`
- Valores: `true`, `false`

Insertar espacio delante de comas

- Nombre: `cpp_space_before_comma`
- Valores: `true`, `false`

Insertar espacio detrás de comas

- Nombre: `cpp_space_after_comma`
- Valores: `true`, `false`

Quitar espacios delante y detrás de operadores de miembro

- Nombre: `cpp_space_remove_around_member_operators`
- Valores: `true`, `false`

Insertar espacio delante de dos puntos para base en declaraciones de tipos

- Nombre: `cpp_space_before_inheritance_colon`
- Valores: `true`, `false`

Insertar espacio delante de dos puntos para los constructores

- Nombre: `cpp_space_before_constructor_colon`
- Valores: `true`, `false`

Quitar espacio delante de punto y coma

- Nombre: `cpp_space_remove_before_semicolon`
- Valores: `true`, `false`

Insertar espacio tras signos de punto y coma

- Nombre: `cpp_space_after_semicolon`
- Valores: `true`, `false`

Quitar espacios entre operadores unarios y sus operandos

- Nombre: `cpp_space_remove_around_unary_operator`
- Valores: `true`, `false`

Espaciado para operadores binarios

- Nombre: `cpp_space_around_binary_operator`
- Valores:
 - `insert`. Insertar espacios delante y detrás de operadores binarios.
 - `remove`. Quitar espacios alrededor de operadores binarios.
 - `ignore`. No cambiar espacios alrededor de los operadores binarios.

Espaciado para operadores de asignación

- Nombre: `cpp_space_around_assignment_operator`
- Valores:
 - `insert`. Insertar espacios alrededor de los operadores de asignación.
 - `remove`. Quitar espacios alrededor de los operadores de asignación.
 - `ignore`. No cambiar espacios alrededor de los operadores de asignación.

Alineación de puntero o referencia

- Nombre: `cpp_space_pointer_reference_alignment`
- Valores:
 - `left`. Alinear a la izquierda.
 - `center`. Alinear al centro.
 - `right`. Alinear a la derecha.
 - `ignore`. Dejar sin cambios.

Espaciado para operadores condicionales

- Nombre: `cpp_space_around_ternary_operator`
- Valores:
 - `insert`. Insertar espacios alrededor de los operadores condicionales.
 - `remove`. Quitar espacios alrededor de los operadores condicionales.
 - `ignore`. No cambiar espacios alrededor de los operadores condicionales.

Opciones de ajuste

Opciones de ajuste para bloques

- Nombre: `cpp_wrap_preserve_blocks`
- Valores:
 - `one liners`. No ajustar bloques de código de una línea.

- `one_liners`. No ajustar bloques de código de una línea.
- `all_one_line_scopes`. No ajustar bloques de código donde haya llaves de apertura y cierre en la línea siguiente.
- `never`. Aplicar siempre la configuración de Nuevas líneas para los bloques.

Vea también

- [EditorConfig.org](https://editorconfig.org/)
- [Compatibilidad de EditorConfig con un servicio de lenguaje](#)
- [Características del editor de código](#)