Artistic Style 1.15.3 (<http://www.bigfoot.com/~davidsont/astyle>)

(created by Tal Davidson, [davidsont@bigfoot.com](mailto:davidsont@bigfoot.com))

Usage : astyle [options] < Original > Beautified

astyle [options] Foo.cpp Bar.cpp [...]

When indenting a specific file, the resulting indented file RETAINS the original file-name. The original pre-indented file is renamed, with a suffix of “.orig” added to the original filename.

By default, astyle is set up to indent C/C++/C# files, with 4 spaces per indent, a maximal indentation of 40 spaces inside continuous statements, and NO formatting.

# Option’s Format

Long options (starting with ‘—‘) must be written one at a time.

Short options (starting with ‘-‘) may be appended together.

Thus, -bps4 is the same as -b -p -s4.

# Predefined Styling options

* style=ansi

ANSI style formatting/indenting.

* style=kr

Kernighan&Ritchie style formatting/indenting.

* style=gnu

GNU style formatting/indenting.

* style=java

Java mode, with standard java style formatting/indenting.

* style=linux

Linux mode (i.e. 8 spaces per indent, break definition-block brackets but attach command-block brackets.

# Indentation options

* c OR --mode=c

Indent a C, C++ or C# source file (default)

* j OR --mode=java

Indent a Java™ source file

* s OR -s# OR --indent=spaces=#

Indent using # spaces per indent. Not specifying # will result in a default of 4 spaces per indent.

* t OR -t# OR --indent=tab=#

Indent using tab characters, assuming that each tab is # spaces long. Not specifying # will result in a default assumption of 4 spaces per tab.

* T# OR --force-indent=tab=# Indent using tab characters, assuming that each tab is # spaces long. Force tabs to be used in areas Astyle would prefer to use spaces.
* C OR --indent-classes

Indent ‘class’ blocks, so that the inner ‘public:’, ‘protected:’ and ‘private: headers are indented in relation to the class block.

* S OR --indent-switches

Indent ‘switch’ blocks, so that the inner ‘case XXX:’ headers are indented in relation to the switch block.

* K OR --indent-cases

Indent ‘case XXX:’ lines, so that they are flush with their bodies..

* N OR --indent-namespaces

Indent the contents of namespace blocks.

* B OR --indent-brackets

Add extra indentation to ‘{’ and ‘}’ block brackets.

* G OR --indent-blocks

Add extra indentation entire blocks (including brackets).

* L OR --indent-labels

Indent labels so that they appear one indent less than the current indentation level, rather than being flushed completely to the left (which is the default).

* m# OR --min-conditional-indent=#

Indent a minimal # spaces in a continuous conditional belonging to a conditional header.

* M# OR --max-instatement-indent=#

Indent a maximal # spaces in a continuous statement, relatively to the previous line.

* E OR --fill-empty-lines

Fill empty lines with the white space of their previous lines.

* indent-preprocessor

Indent multi-line #define statements

# Formatting options

* b OR --brackets=break

Break brackets from pre-block code (i.e. ANSI C/C++ style).

* a OR --brackets=attach

Attach brackets to pre-block code (i.e. Java/K&R style).

* l OR --brackets=linux

Break definition-block brackets and attach command-block brackets.

* brackets=break-closing-headers

Break brackets before closing headers (e.g. ‘else’, ‘catch’, ..).

Should be appended to—brackets=attach or—brackets=linux.

* o OR --one-line=keep-statements

Don’t break lines containing multiple statements into multiple single-statement lines.

* O OR --one-line=keep-blocks

Don’t break blocks residing completely on one line

* p OR --pad=oper

Insert space paddings around operators only.

* pad=paren

Insert space paddings around parenthesies only.

* P OR --pad=all

Insert space paddings around operators AND parenthesis.

* convert-tabs

Convert tabs to spaces.

* break-blocks

Insert empty lines around unrelated blocks, labels, classes, ...

* break-blocks=all

Like—break-blocks, except also insert empty lines around closing headers (e.g. ‘else’, ‘catch’, ...).

* break-elseifs

Break ‘else if()’ statements into two different lines.

# Other options:

* suffix=####

Append the suffix #### instead of ‘.orig’ to original filename.

* X OR --errors-to-standard-output

Print errors and help information to standard-output rather than to standard-error.

* v OR --version

Print version number

* h OR -? OR --help

Print this help message

# Default options file:

Artistic Style looks for a default options file in the following order:

1. The contents of the ARTISTIC\_STYLE\_OPTIONS environment variable if it exists.

2. The file called .astylerc in the directory pointed to by the HOME environment variable ( i.e. $HOME/.astylerc ).

3. The file called .astylerc in the directory pointed to by the HOMEPATH environment variable ( i.e. %HOMEPATH%\.astylerc ). If a default options file is found, the options in this file will be parsed BEFORE the command-line options. Options within the default option file may be written without the preliminary ‘-‘ or ‘—‘.