Qt Creator Manual > Analyzing Code with Cppcheck

Analyzing Code with Cppcheck

Cppcheck is a static analysis tool that detects errors in C++ code. Static analysis is performed on the source code without actually executing the application.

The experimental Cppcheck Diagnostics plugin integrates diagnostics that are generated by the Cppcheck tool into the C++ editor.

Cppcheck is automatically run on open files. To select the files to check in the currently active project, select **Analyze** > **Cppcheck**.

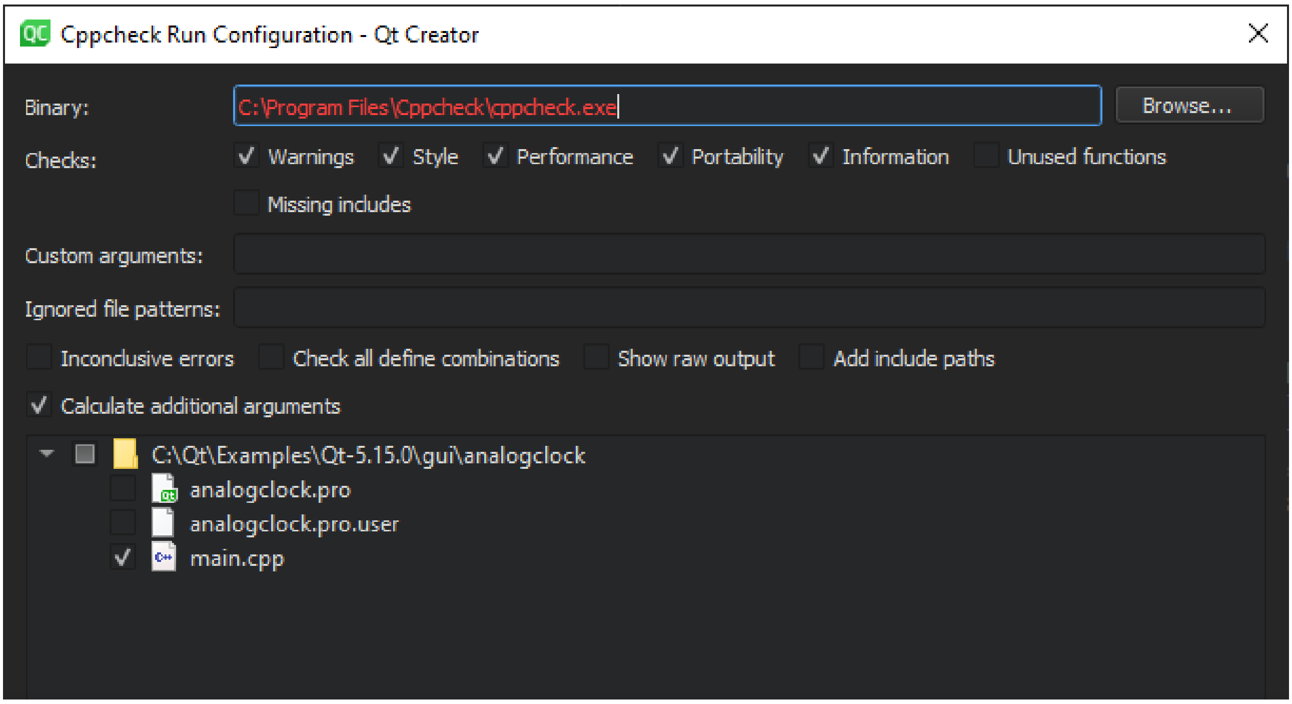
Enabling the Cppcheck Plugin

To enable the Cppcheck plugin:

1. Select **Help** > **About Plugins** > **Code Analyzer** > **Cppcheck** to enable the plugin.
2. Select **Restart Now** to restart Qt Creator and load the plugin.

Running Cppcheck on Selected Files

1. Select **Analyze** > **Cppcheck**.



Cppcheck Run Configuration - Qt Creator

Binary: C:\Program Files\Cppcheck\cppcheck.exe

Checks: ☑ Warnings ☑ Style ☑ Performance ☑ Portability ☑ Information ☐ Unused functions
☐ Missing includes

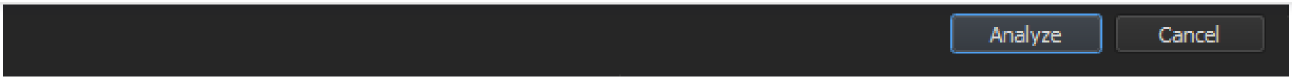
Custom arguments:

Ignored file patterns:

☐ Inconclusive errors ☐ Check all define combinations ☐ Show raw output ☐ Add include paths

☑ Calculate additional arguments

▾ ■ C:\Qt\Examples\Qt-5.15.0\gui\analogclock
- ☐ analogclock.pro
- ☐ analogclock.pro.user
- ☑ main.cpp



2. In the **Binary** field, enter the path to the Cppcheck executable file.
3. In the **Checks** group, select the checks to perform.

> **Note:** By default, Cppcheck uses multiple threads to perform checks. Selecting the **Unused functions** option disables the default behavior.

4. In the **Custom arguments** field, enter additional arguments for running Cppcheck. The arguments might be shadowed by automatically generated ones. To avoid possible conflicts in configuration, select the **Show raw output** check box to see the final arguments.
5. In the **Ignored file patterns** field, enter a filter for ignoring files that match the pattern (wildcard). You can enter multiple patterns separated by commas. Even though Cppcheck is not run on files that match the provided patterns, they might be implicitly checked if other files include them.
6. Select the **Inconclusive errors** check box to also mark possible false positives.
7. Select the **Check all define combinations** check box to check all define combinations. Enabling this option can significantly slow down analysis, but might help to find more issues.
8. Select the **Add include paths** check box to pass the current project's include paths to Cppcheck. Enabling this option slows down checks on big projects, but can help Cppcheck to find missing includes.
9. Select the **Calculate additional arguments** check box to calculate additional arguments based on current project's settings (such as the language used and standard version) and pass them to Cppcheck.
10. Select the files to run Cppcheck on.
11. Select **Analyze**.

Qt Creator runs Cppcheck on the selected files and displays results via text marks or annotations.

To specify the settings above for the automatically run checks, select **Edit** > **Preferences** > **Analyzer** > **Cppcheck**.

‹ Analyzing CPU Usage

Visualizing Chrome Trace Events ›