

[Qt创作者手册](#) > [使用 Cppcheck 分析代码](#)

使用 Cppcheck 分析代码

Cppcheck是一个静态分析工具，用于检测C++代码中的错误。静态分析是在源代码上执行的，而不实际执行应用程序。

实验性的 Cppcheck 诊断插件将 Cppcheck 工具生成的诊断集成到C++编辑器中。

Cppcheck会自动在打开的文件上运行。若要选择要签入当前活动项目的文件，请选择“**分析**>Cppcheck”。

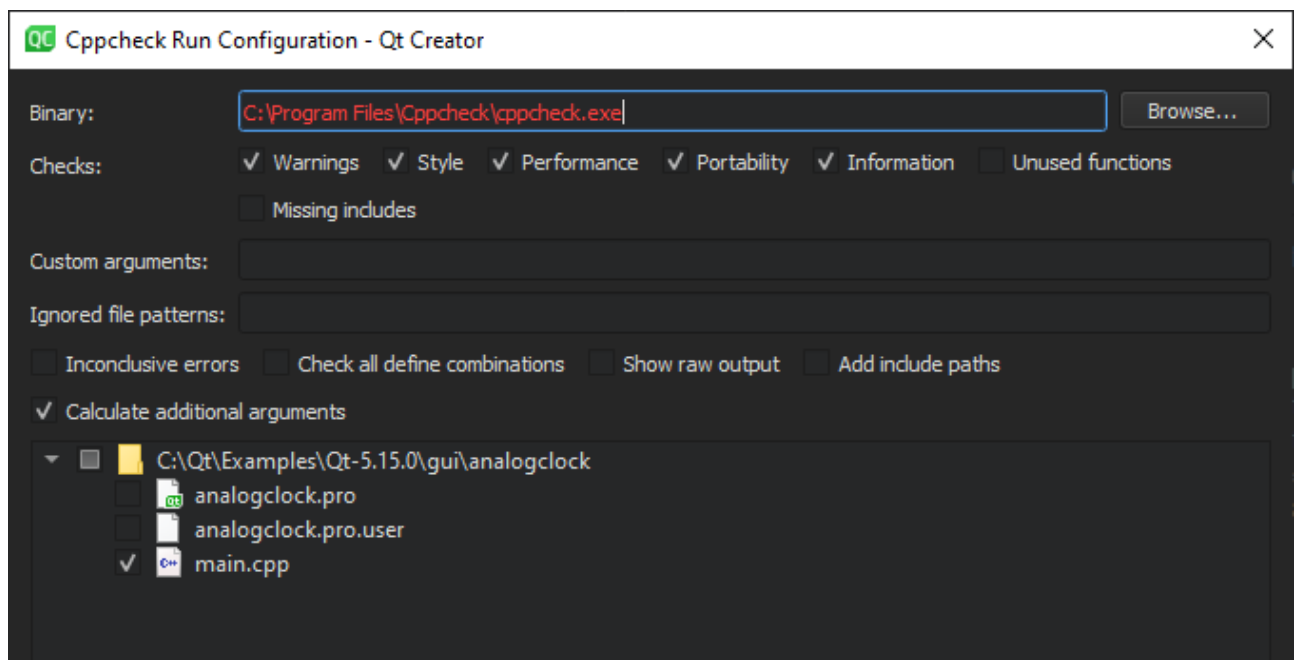
启用 Cppcheck 插件

[Qt创建者手册8.0.2](#)
[Topics >](#)

2. 选择**立即重启以重启**Qt Creator并加载插件。

对选定文件运行 Cppcheck

1. 选择“**分析**>Cppcheck”。



Analyze

Cancel

2. 在**二进制**字段中，输入 Cppcheck 可执行文件的路径。
3. 在“检查”组中，选择要执行的**检查**。

注意：默认情况下，Cppcheck 使用多个线程来执行检查。选择**“未使用的函数”**选项将禁用默认行为。

4. 在自定义**参数**字段中，输入用于运行 Cppcheck 的其他参数。参数可能会被自动生成的参数遮蔽。若要避免配置中可能出现的冲突，请选中**“显示原始输出”**复选框以查看最终参数。
5. 在 **忽略的文件模式** 字段中，输入用于**忽略与模式匹配的文件**（通配符）的筛选器。您可以输入多个模式，以逗号分隔。即使 Cppcheck 未在与提供的模式匹配的文件上运行，如果其他文件包含它们，也可能隐式检查它们。
6. 选中**“不确定错误”**复选框以同时标记可能的误报。
7. 选中**“选中所有定义组合”**复选框以选中**所有定义组合**。启用此选项可能会显著减慢分析速度，但可能有助于发现更多问题。
8. 选中**“添加包含路径”**复选框，将当前项目的**包含路径**传递给 Cppcheck。启用此选项会减慢对大型项目的检查速度，但可以帮助 Cppcheck 查找缺少的包含。
9. 选中**“计算其他参数”**复选框以根据当前项目的设置（例如使用的语言和标准版本）计算**其他参数**，并将其传递给 Cppcheck。
10. 选择要对其运行 Cppcheck 的文件。
11. 选择**“分析”**。

Qt Creator对所选文件运行Cppcheck，并通过文本标记或注释显示结果。

若要为自动运行检查指定上述设置，请选择**“编辑>首选项>分析器>Cppcheck”**。

[< 分析 CPU 使用率](#)[可视化 Chrome 跟踪事件 >](#)

©2022 Qt有限公司 此处包含的文档贡献的版权归 他们各自的所有者。此处提供的文档根据自由软件基金会发布的GNU 自由文档许可证版本 1.3的条款进行许可。Qt和相应的徽标是Qt有限公司在芬兰和/或全球其他国家的商标。所有其他商标均为财产 其各自所有者。

[联系我们](#)[公司](#)[发牌](#)[关于我们](#)[条款和条件](#)

职业
办公地点

支持

支持服务
专业服务
合作 伙伴
训练

对于客户

支持中心
下载
Qt登录
联系我们
客户成功案例

社区

为Qt做贡献
论坛
维基
下载
市场

© 2022 Qt公司

[反馈](#) [登录](#)