



Qt 创建者手册 8.0.2
Topics >

🔍 搜索

Qt 创建者手册 > [分析代码](#)

可以在**调试**模式下使用代码分析工具。若要切换到**“调试”**模式，请在模式选择器中选择**“调试”**，或选择**“分析”**菜单，然后选择一个工具。处于**调试**模式时，可以通过在工具栏上的菜单中选择工具来切换工具。

可以将**“调试”**模式下的视图拖放到屏幕上的新位置。将保存视图的大小和位置以供将来的会话使用。选择**“视图”>“视图”>“重置为默认布局”**，将视图重置为其原始大小和位置。

可以在**调试**模式下使用以下代码分析工具：

› 秦淮公路探查器

在运行 QML 代码时检查绑定评估、信号处理和绘制操作。这对于识别潜在的瓶颈非常有用，尤其是在评估绑定时。

› 可可

例如，分析应用程序作为测试套件的一部分的运行方式，并使用结果使测试更加高效和完整。

› 瓦尔格林代码分析工具

使用 Memcheck 工具检测内存管理中的问题，并使用 Callgrind 工具查找代码中的缓存未命中。

› 叮当工具

通过使用 C-Tidy 和 Clazy 检测 C、C++ 和目标 C 程序中的问题。

› 赫布

使用 Windows 上的 Heob 堆观察器来检测缓冲区溢出和内存泄漏。

› 性能分析器

使用集成了 Linux Perf 工具的性能分析器分析嵌入式应用程序和 Linux 桌面应用程序的 CPU 使用情况。

› 断续器

使用实验性的 Cppcheck 插件来检测未定义的行为和危险的编码结构。

› 铬跟踪格式可视化工具

使用浏览器跟踪格式（CTF）可视化工具查看 Chrome 跟踪事件。这在查看难以使用内置跟踪查看器（）可视化的大型跟踪文件时特别有用。`chrome://tracing`

‹ 调试器疑难解答

分析 QML 应用程序 ›

©2022 Qt Ltd. 此处包含的文档贡献是其各自所有者的版权。此处提供的文档是根据自由软件基金会发布的 [GNU 自由文档许可证 1.3 版](#) 的条款进行许可的。Qt 及其相应的徽标是 Qt 有限公司在芬兰和/或全球其他国家的商标。所有其他商标均为其各自所有者的财产。



联系我们

公司

关于我们
投资者
编辑部
职业
办公地点

支持

支持服务
专业服务
合作 伙伴
训练

社区

为Qt做贡献
论坛
维基
下载
市场

发牌

条款及细则
开源
常见问题

对于客户

支持中心
下载
秦特登录
联系我们
客户成功案例