

[Qt创作者手册](#) > [使用 Memcheck 检测内存泄漏](#)

使用 Memcheck 检测内存泄漏



您可以使用 Valgrind 工具套件中包含的 Memcheck 工具来检测与应用程序中的内存管理相关的问题。您可以将该工具与 GDB 调试器一起使用。检测到问题时，应用程序将中断，您可以对其进行调试。

注意：您可以在 Linux 上本地安装和运行 Memcheck。您可以从任何开发计算机在远程主机或设备上运行它。在 Windows 上，您可以使用 [Heob](#) 堆观察器来接收类似的结果。

下载并安装 Valgrind 工具后，您可以使用 Qt Creator 中的 Memcheck。

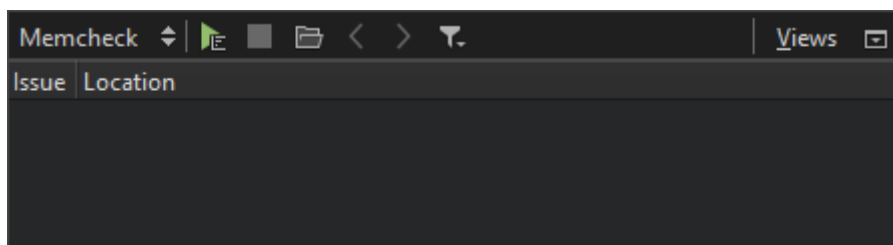
要分析应用程序：

[Qt 创建者手册 8.0.2](#)
Topics >


2. 选择“调试”以打开“调试”模式，然后在工具栏上选择“Memcheck”。
3. 选择按钮  以启动应用程序。
4. 使用应用程序对其进行分析。
5. 选择该  按钮可在“分析”视图中查看分析结果。

当应用程序运行时，Memcheck 会检查内存的所有读写，并拦截分配或释放内存或创建或删除内存块的调用。当您停止 Memcheck 时，将显示结果。单击一行可查看内存泄漏发生的位置以及显示导致该泄漏的原因的堆栈跟踪。

作为收集数据的替代方法，您可以选择  将 XML 格式的外部日志文件加载到 Memcheck 视图中。



将鼠标移到一行上可查看有关函数的详细信息。

若要在行之间移动，请选择  或 。

若要筛选结果，请选择 ，然后选择  要在视图中显示的问题类型。您可以查看和隐藏明确和可能的内存泄漏、

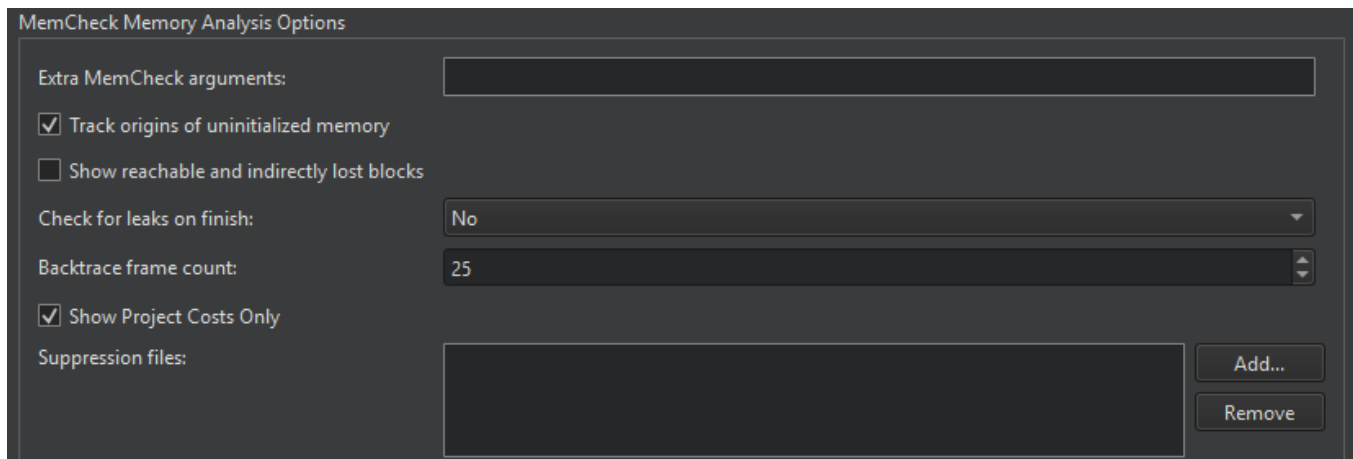
选择内存分析选项

可以为所有项目全局指定分析器设置，也可以在项目的运行设置中为每个项目单独指定分析器设置。

要指定 Valgrind 的全局设置，请选择**编辑>首选项>分析器**。**内存检查内存分析**选项组包含内存检查选项。

在Extra Memcheck 参数中，指定用于启动可执行文件的其他参数。

堆栈跟踪可能会变得非常大且令人困惑，因此，自下而上读取它们会有所帮助。如果堆栈跟踪不够大或太大，请选择**编辑>首选项>分析器**，并在**回溯跟踪帧计数**字段中定义堆栈跟踪的长度。



Memcheck还报告未初始化值的使用，最常见的消息是**条件跳转或移动取决于未初始化的值**。若要确定这些错误的根本原因，默认情况下会选中**跟踪未初始化内存的来源**复选框。您可以取消选择它以使 Memcheck 运行得更快。

Memcheck 在客户端应用程序完成时搜索内存泄漏。若要查看发生的泄漏量，请在**完工时检查泄漏**字段中选择**仅摘要**。若要同时查看每个泄漏的详细信息，请选择**完整**。

显示可访问和间接丢失的块

可访问块是由指针或指针链指向的块，这些块可能在应用程序退出之前已释放。**间接丢失的块**被视为丢失，因为指向它们的所有块本身都丢失了。例如，丢失的根节点的所有子节点都将间接丢失。

默认情况下，Memcheck 不会报告可访问和间接丢失的块。要报告它们，请选择**显示可访问和间接丢失的块**。

抑制错误

Memcheck检测到系统库中的许多问题，例如操作系统预装的C库。由于您无法轻松修复它们，因此要抑制它们。Valgrind 读取要在启动时抑制的错误列表。默认抑制文件由 ./configure 脚本在构建系统时创建。

如果项目的某些部分包含无法修复的错误，并且您不希望收到提醒，则可以编写自己的抑制文件。单击**内存检查分析**对话框中的**添加**以**添加抑制文件**。有关写入抑制文件的详细信息，请参阅Valgrind文档中的抑制错误。

[< 使用瓦尔格林德代码分析工具](#)

[性能分析函数执行 >](#)

©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

- 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

- 条款和条件
- 开源
- 常见问题

对于客户

- 支持中心
- 下载
- Qt登录
- 联系我们
- 客户成功案例