

[Qt创作者手册](#) > [使用clang代码模型解析C++文件](#)

使用 clang 代码模型解析C++文件

代码模型是 IDE 的一部分，它理解用于编写应用程序的语言。它是允许Qt Creator提供以下服务的框架：

- › 代码完成
- › 句法和语义突出显示
- › 使用[定位器](#)、[跟随符号](#)等在代码中导航
- › 使用[类浏览器](#)、[大纲](#)等检查代码
- › 诊断
- › 工具提示

Qt创建者手册8.0.2
Topics >

Qt Creator带有一个插件，可以在Clangd之上为C++提供其中一些服务。

关于 Clang 代码模型

Clang 项目提供了用于解析 C 语言系列源文件的库。您通过警告和错误标记获得的反馈与编译器为您提供的反馈相同，而不是像使用内置的Qt Creator代码模型时那样不完整的集合或近似值。Clang 专注于诊断的详细信息，例如，如果代码包含拼写错误，这将非常有用。我们通过实现LSP服务器的 clangd 工具使用这些库。

Clang跟上C++语言的发展。在撰写本文时，它支持 C++98/03、C++11、C++14、C++17、C89、C99、Objective-C 和 Objective-C++。

不利的一面是，对于使用Clang作为代码模型的大型项目，使用内置代码模型要慢。Clang 不需要生成目标文件，但仍需要解析和分析源文件。对于仅使用 STL 的小型项目，这相对较快。但对于包含多个文件的大型项目，处理单个文件和所有包含的文件可能需要一段时间。

Clang 代码模型插件现在提供了以前由内置 C/C++ 代码模型提供的一些服务。目前，已实现以下服务：

- › 代码完成
- › 句法和语义突出显示
- › 诊断
- › 符号概述
- › 工具提示

› 查找符号的出现次数

若要改用内置代码模型，请选择“**编辑>首选项>C++>clangd**”，然后取消选中“**使用 clangd**”复选框。此设置也存在于项目级别，因此您可以通常启用基于 clang 的服务，但对于某些项目将其关闭，反之亦然。

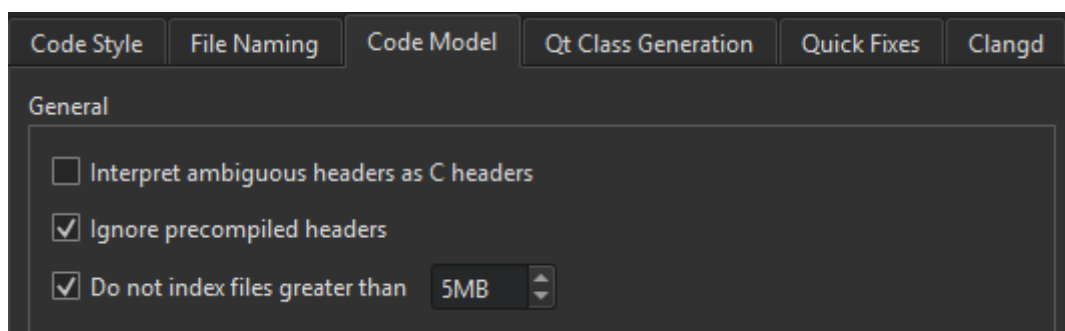
您可以全局或单独配置Clang诊断：

- › Clang 代码模型（全局或项目级别）
- › Clang工具（全球或项目级别）

配置 Clang 代码模型

要全局配置 Clang 代码模型，请执行以下操作：

1. 选择“**编辑>首选项>C++>代码模型**”。



2. 如果主要使用 C 进行开发，则指示代码模型将不明确的头文件解释为 C 语言文件，请选中“**将不明确的头文件解释为 C 标头**”复选框。
3. 要处理预编译标头，请取消选中“**忽略预编译标头**”复选框。
4. 为避免因索引通常由脚本或代码自动生成的大型源文件而导致的内存不足崩溃，默认情况下，要编制索引的文件大小限制为 5MB。若要调整限制，请编辑“**不索引大于以下值的文件**”复选框的值。要为所有文件编制索引，请取消选中该复选框。

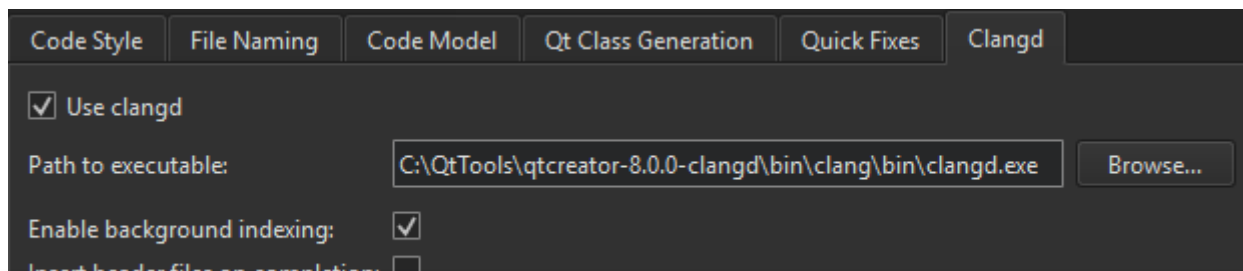
配置叮当声

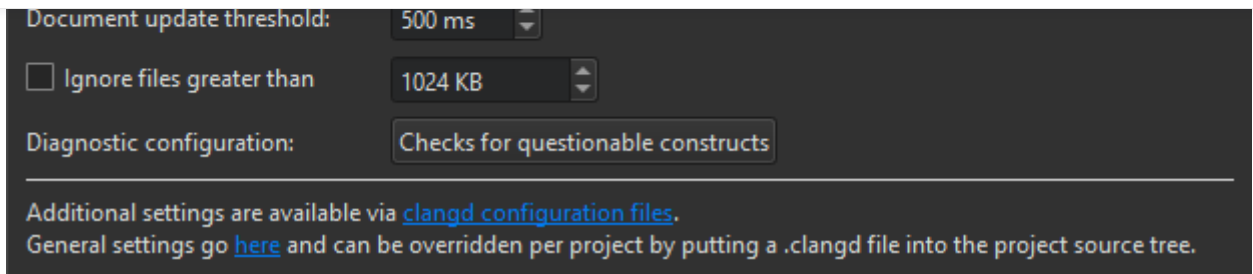
clangd索引为查找引用、跟踪光标下的符号以及使用定位器等服务提供准确和完整的结果，即使对于复杂的构造也是如此。当您**打开项目**时，clangd 会扫描源文件以生成索引。对于大型项目，这可能需要一段时间，但索引是持久的，重新扫描是增量的，因此关闭并重新启动Qt Creator不会丢失任何内容。

大纲视图中的文档**大纲**由 clang 的文档符号支持，这使得结果比以前更可靠。

要指定 clangd 的设置，请执行以下操作：

1. 选择**编辑>首选项>C++>叮叮当当>使用叮叮当当**。





2. 在“**可执行文件的路径**”中，输入指向 clangd 版本 13 或更高版本的路径。
3. 要在全局符号搜索期间获得更准确的结果，请选择**启用后台索引**。但是，这会增加首次打开项目时的 CPU 负载。
4. 选择“**完成时插入头文件**”以允许 clangd 在符号完成过程中插入头文件。
5. 默认情况下，clangd 会尝试使用所有未使用的核心。您可以设置要在**工作线程计数**中使用的固定内核数。
6. 在**文档更新阈值**中，指定Qt Creator在将文档更改发送到服务器之前等待的时间。如果文档在等待期间再次更改，则会重置此超时。
7. 选择“**忽略大于**”的文件，通过忽略大文件来加快分析速度。在复选框旁边的字段中指定要分析的文件的最大大小。
8. “**诊断配置**”字段显示要执行的 Clang 检查。单击字段的值以打开“**诊断配置**”对话框，您可以在其中选择和编辑要执行的检查。

叮当检查

除了使用内置检查之外，还可以选择“**复制**”以创建它们的副本，并根据需要编辑副本。

生成系统警告**显示生成系统指定的警告**。

对可疑结构的检查结合了对易于避免的**可疑结构**和一些其他问题的检查。-Wall-Wextra

叮当检查开始。每个检查还有一个以开头的否定版本。-W-Wno

请记住，某些选项会打开其他选项。有关详细信息，请参阅请求或**禁止显示警告的选项**或GCC 或 Clang 手册页。

在项目级别指定 Clang 代码模型设置

可以通过选择“项目>clangd”在**项目**级别指定 Clang 代码模型设置。

使用编译数据库

JSON 编译数据库格式指定如何独立于生成系统重播单个生成。

编译数据库基本上是文件列表和用于编译文件的编译器标志。数据库用于向代码模型提供必要的信息，以便在打开文件进行编辑时正确分析代码。

若要根据代码模型具有的信息生成编译数据库，请选择“**生成**>**生成编译数据库**”。

可以将文件（如非 C 文件）添加到`compile_database.json.files`中的项目中。

您可以使用实验性编译数据库项目管理器打开编译数据库中的文件，并访问 Clang 代码模型提供的所有编辑功能。

可以在“**项目**”模式下为编译数据库项目指定自定义生成步骤和运行设置。有关更多信息，请参见[添加自定义生成步骤](#)和[指定运行设置](#)。

要启用插件，请选择**帮助>关于插件>构建系统>编译数据库项目经理**”。然后选择**立即重启以重启**Qt Creator并加载插件。

< 比较文件

发现 >

©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

发牌

- 条款和条件
- 开源
- 常见问题

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



