

Qt创作者手册 > [使用语言服务器](#)

使用语言服务器

对于几种编程语言，只要 IDE 支持通过语言服务器协议（LSP）进行通信，就可以使用语言服务器向 IDE 提供有关代码的信息。这使 IDE 能够提供以下服务：

- › 代码完成
- › 向语言服务器发送文档格式请求，以使用“**编辑**>**首选项**>**文本编辑器**>**行为**”中指定的设置自动设置文档格式
- › 突出显示光标下的符号
- › 查看函数工具提示
- › 语义突出显示，如语义突出显示协议扩展提案中所定义
- › 使用定位器在代码中导航或移动到符号定义
- › 通过在“大纲”视图或编辑器工具栏上的“**符号**”列表中查看文档大纲来检查代码
- › 查找对符号的引用
- › 重命名光标下的符号
- › 代码操作
- › 将来自语言服务器的诊断显示为工具提示。还可以选择要显示其诊断的代码范围。

通过为语言服务器协议提供客户端，Qt Creator可以支持除C++之外的其他几种编程语言的上述功能。但是，客户端不支持需要特殊处理的语言服务器。

为语言服务器添加 MIME 类型

Qt Creator使用文件的MIME类型来确定当您打开文件进行编辑时从哪个语言服务器请求信息。添加新的 MIME 类型或文件模式以匹配语言服务器。如果未设置至少一个 MIME 类型或文件模式，则不会将任何文件发送到语言服务器。这样做是为了避免不必要的流量和不准确的信息，因为只有当已知文件由语言服务器处理时，它们才会发送到语言服务器。有关Qt Creator如何使用MIME类型的更多信息，请参阅[编辑MIME类型](#)。

指定语言客户端的设置

Qt Creator支持为Android开发添加Java语言服务器。默认情况下会添加 Python 语言服务器，您可以编辑其首选项。对于其他语言，可以添加通用 stdIO 语言服务器。

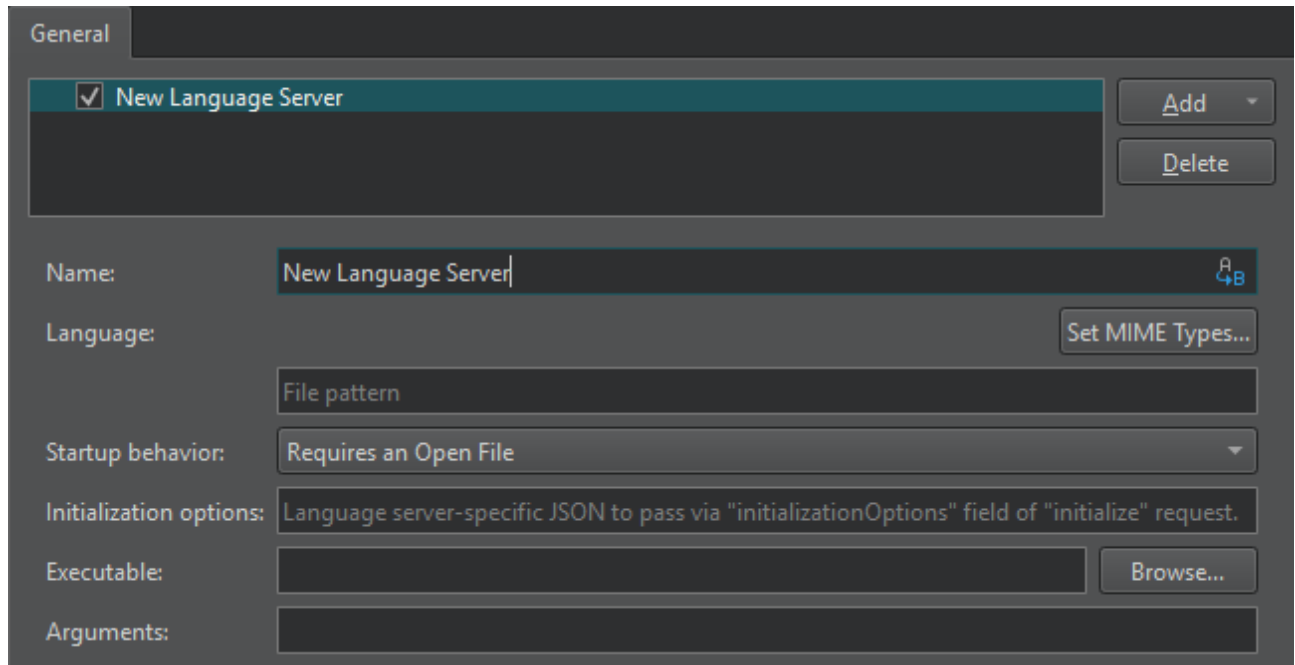
要添加语言服务器，请在macOS上选择**编辑**>**首选项**>**语言客户端**>Add（或Qt Creator>Preferences>Language


若要从列表中删除语言服务器，请选择“删除”。

通用标准IO语言服务器

添加通用语言服务器：

1. 选择“编辑>首选项>语言客户端>添加>通用 StdIO 语言服务器”以添加通用语言服务器。

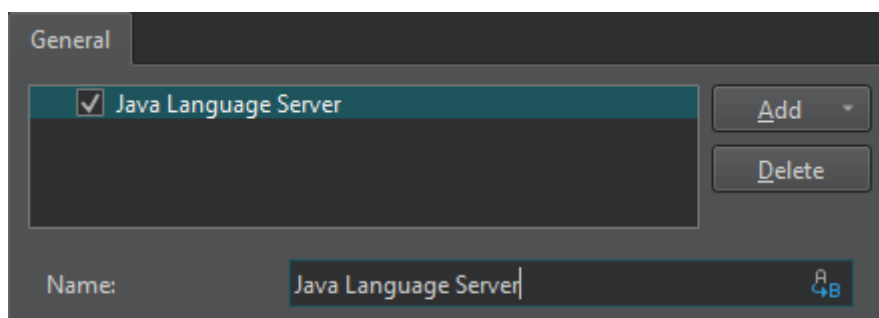



2. 在“名称”字段中，输入语言服务器的名称。选择“（变量）”按钮以使用变量作为服务器名称。有关更多信息，请参阅[使用 Qt 创建者变量](#)。
3. 在“语言”字段中，选择“设置 MIME 类型”以选择要发送到语言服务器的文件的 MIME 类型。在下面的字段中，您可以输入文件模式以扩展 MIME 类型，用分号分隔。
4. 在“启动行为”字段中，选择语言服务器是在 Qt Creator 启动时启动，还是在打开具有匹配 MIME 类型的项目或文件时启动。“常规消息”显示有关与语言服务器的连接的信息。
5. 在“初始化选项”字段中，可以添加要传递给请求的语言服务器特定 JSON 属性。initialize
6. 在“可执行文件”字段中，输入语言服务器可执行文件的路径。
7. 在“参数”字段中，输入任何必需的命令行参数。选择“变量”以使用变量作为参数。

爪哇语言服务器

要添加 Java 语言服务器，请执行以下操作：

1. 选择编辑>首选项>语言客户端>添加>Java 语言服务器以添加 Java 语言服务器。

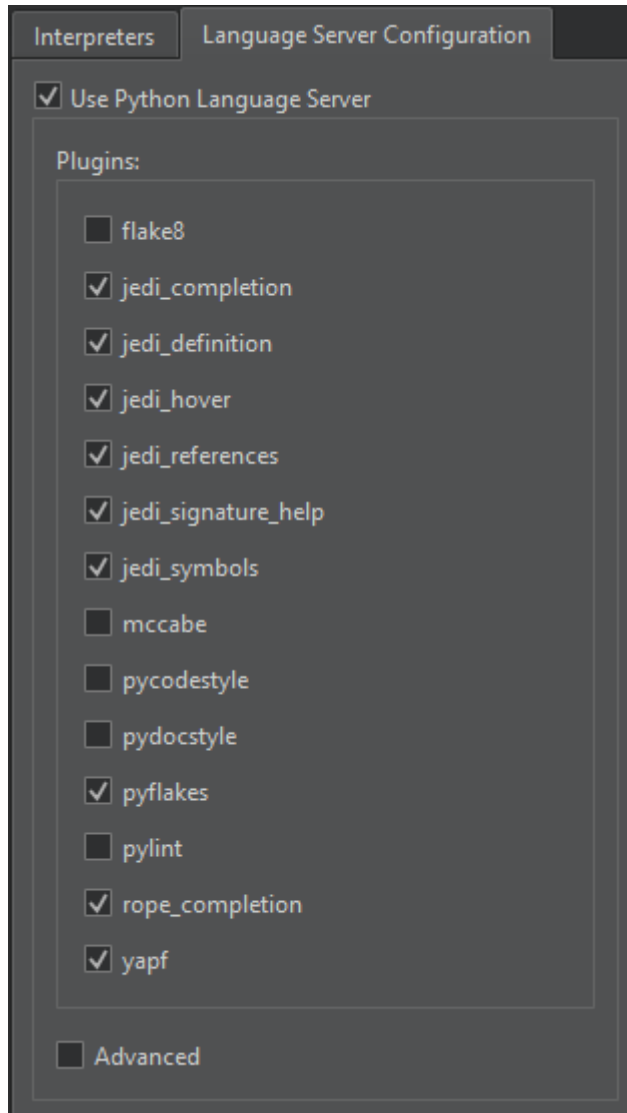


2. 在“名称”字段中，输入语言服务器的名称。选择“ (变量)”按钮以使用**变量**作为服务器名称。有关更多信息，请参阅[使用 Qt 创建者变量](#)。
3. 在“Java”字段中，输入 Java 可执行文件的路径。
4. 在“Java 语言服务器”字段中，输入 Java **语言服务器**文件的路径。 .jar

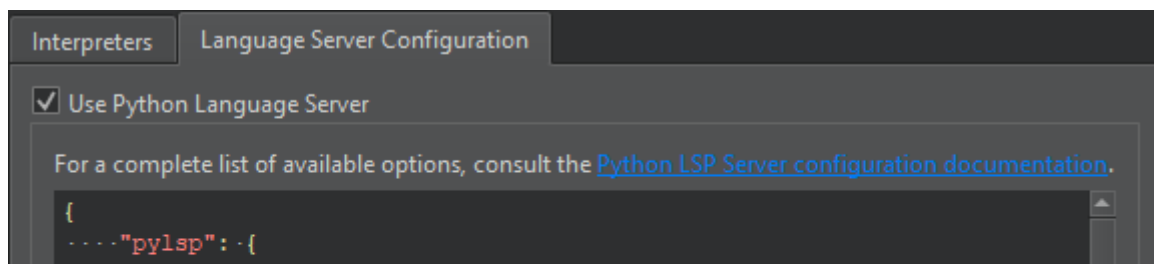
蟒蛇语言服务器

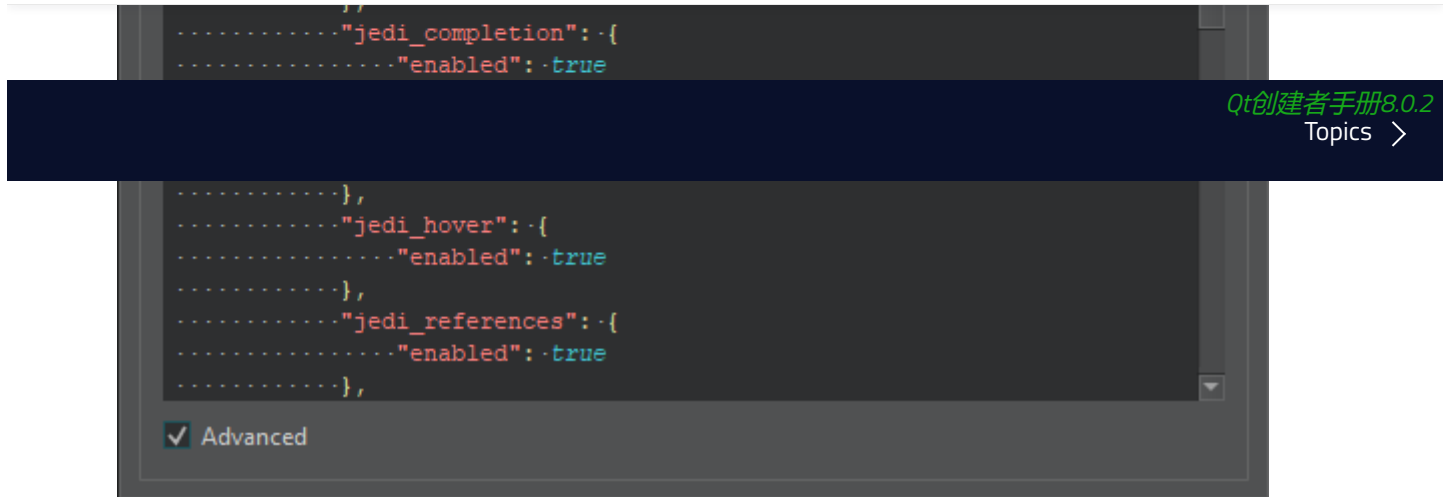
要设置 Python 语言服务器的首选项，请执行以下操作：

1. 选择**编辑>首选项>Python>语言服务器配置**以选择要使用的 Python **语言服务器**插件。



2. 选择“**高级**”以配置插件。





有关配置选项的完整列表，请参阅[Python 语言服务器配置](#)。

要禁用 Python 语言服务器，请取消选择**使用 Python 语言服务器**。

QML 语言服务器

Qt 6.4附带了为QML提供补全和警告的语言服务器。要将其设置为[通用 StdIO 语言服务器](#)，请选择 MIME 类型和可执行文件。qmltext/x-qmlapplication/x-qt.ui+qml<Qt Installation>/bin/qmls

如果语言服务器与插件一起使用，则可能会显示重复的建议和警告。为避免这种情况，请按照[启用和禁用插件](#)中所述禁用编辑器插件。QmlJSEditor

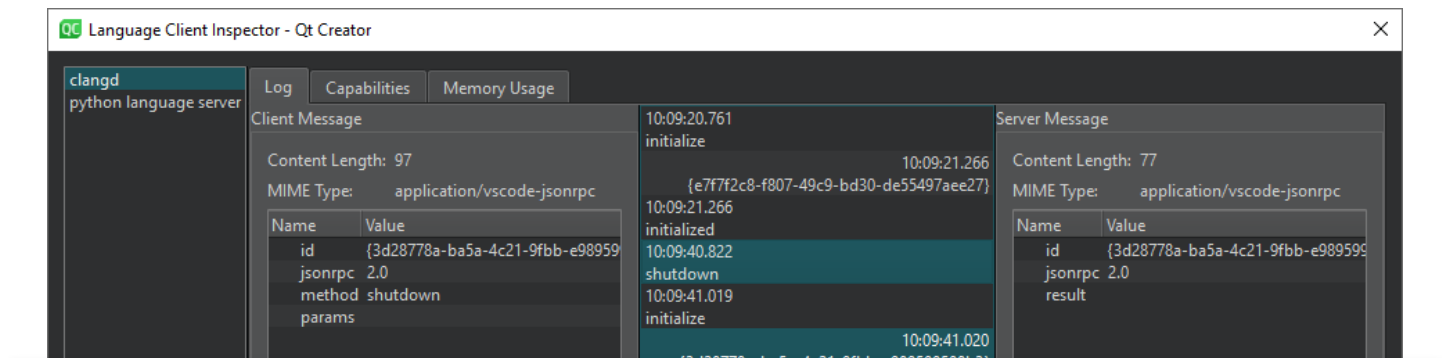
支持的定位器过滤器

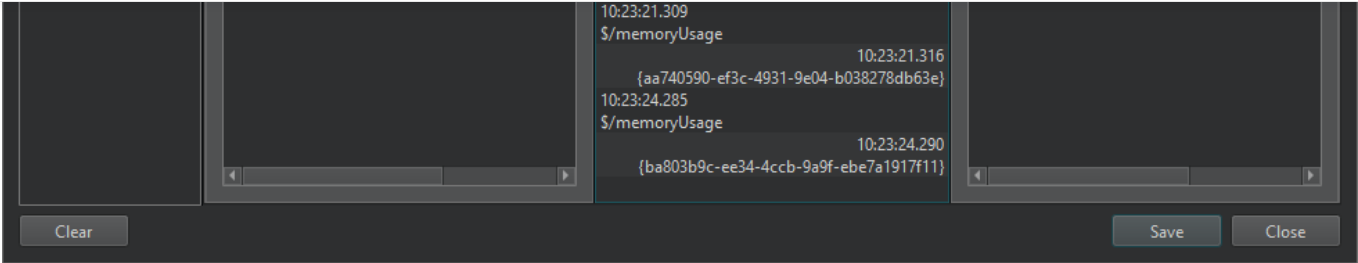
定位器不仅可用于浏览文件，还可用于浏览[定位器过滤器](#)定义的任何项目。语言客户端插件支持以下定位器过滤器：

- › 在当前项目中定位符号 (：)
- › 在当前文档中查找符号 (。)
- › 在项目中查找类 ()、枚举和函数 () 定义cm

检查语言客户端

Qt Creator向语言服务器发送消息（请求），如果语言服务器能够处理请求，则接收包含所请求信息的响应。要检查Qt Creator与语言服务器之间的通信以及查看服务器功能，请选择Tools>Debug Qt Creator>Inspect Language Client。





该对话框显示正在运行的语言服务器的列表。语言服务器首选项中“启动行为”字段的值确定服务器何时启动。显示的信息取决于语言服务器。

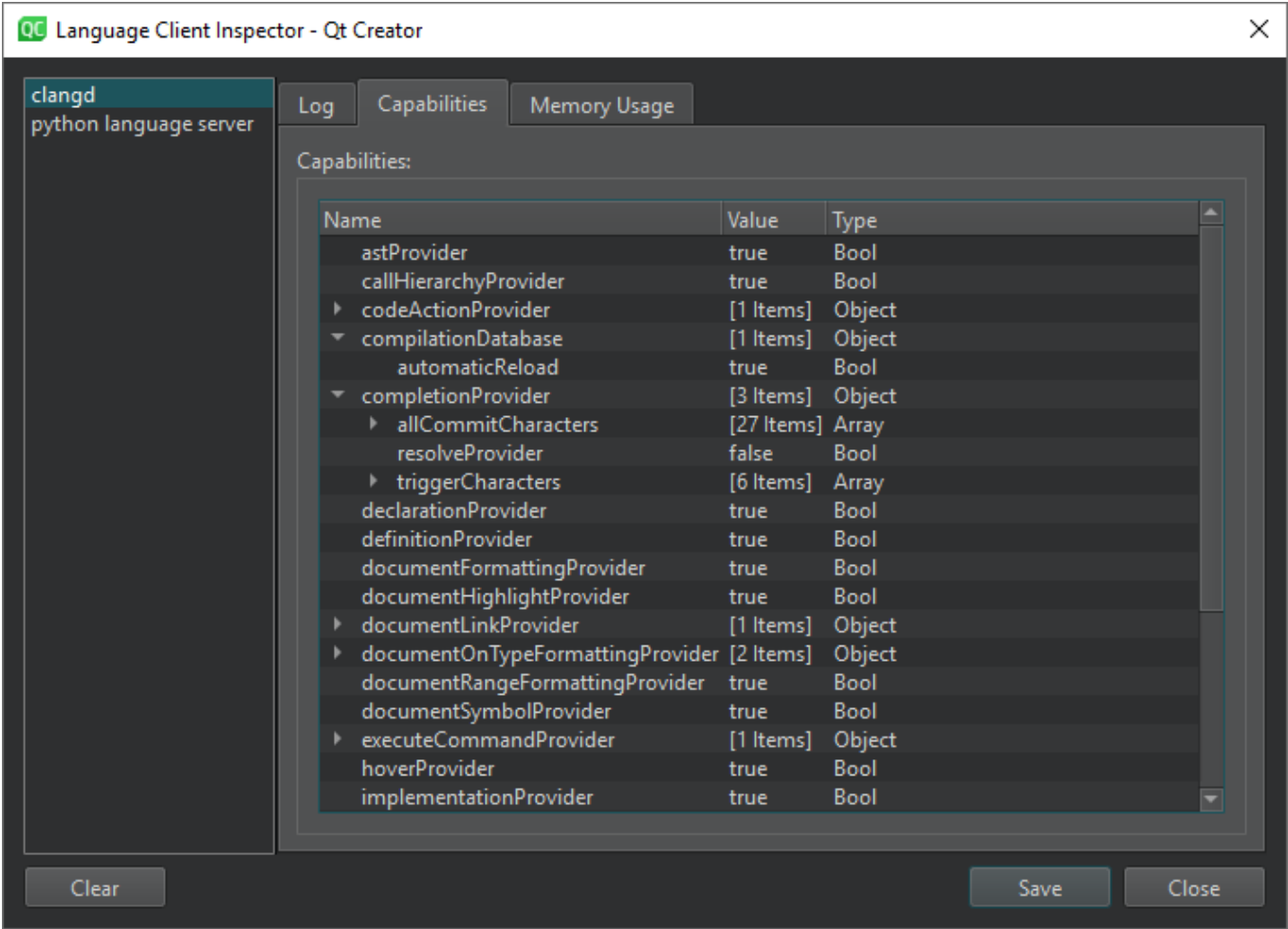
日志显示有关所选日志条目的其他信息。您可以查看客户端消息和服务器消息的内容长度和MIME类型，以及检查Qt Creator和语言服务器之间发送的数据。

若要删除旧条目，请选择“清除”。

能力

在“功能”中，可以检查语言服务器是否能够执行特定任务。您不能在此对话框中修改服务器功能。

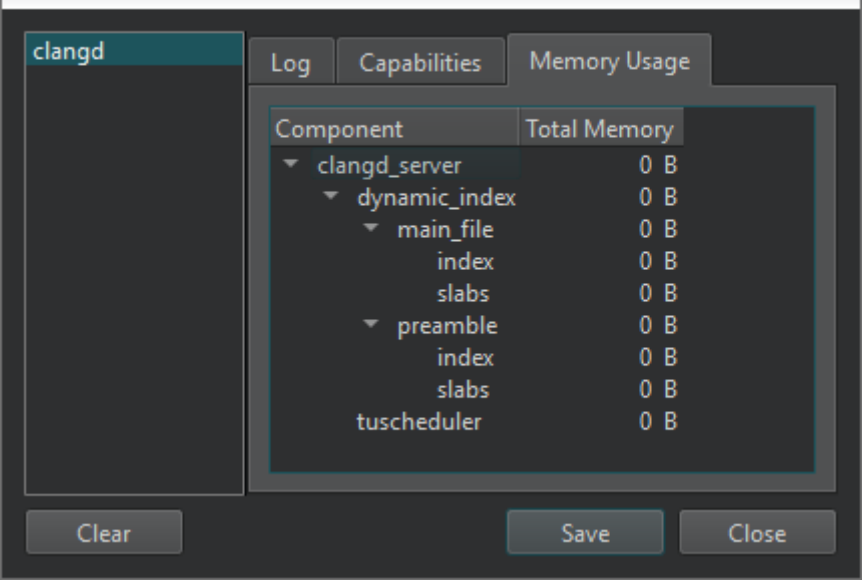
您可以查看功能的名称、值和类型。



对于某些语言服务器，动态功能列出了可用的方法和选项。

内存使用情况

对于 clangd 服务器，您可以在内存使用情况中检查特定组件使用的内存总量。



报告问题

语言服务器客户端主要使用Python和Java进行了测试。如果您在尝试使用它们或其他语言时出现问题，请选择[帮助>报告错误](#)以在[Qt项目错误跟踪器](#)中报告它们。报告应包括带有环境变量集的Qt Creator控制台输出。
`QT_LOGGING_RULES=qtclanguageclient.*=true`

[< 使用假维姆模式](#)

[编辑 MIME 类型 >](#)

©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

主站 对工客白



专业服务
合作 伙伴
训练

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