使用语言服务器

对于几种编程语言，可以使用*语言服务器*向 IDE 提供有关代码的信息，只要它们支持通过语言服务器协议（LSP）进行通信即可。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设置请求，以使用“**编辑**>**首选项**”>**文本编辑器**“中指定的设置自动设置文档格式>**行为**
- 突出显示光标下的符号
- 查看功能工具提示
- 语义突出显示，如语义突出显示协议扩展的提案中所定义
- 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
- 将语言服务器中的诊断信息显示为工具提示。还可以选择要显示诊断信息的代码范围。

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，要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。如果未设置至少一种 MIME 类型或文件模式，则不会向语言服务器发送任何文件。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 languge 服务器处理时，它们才会发送到该服务器。有关 Qt 创建器如何使用 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创建者支持为安卓开发添加一个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。

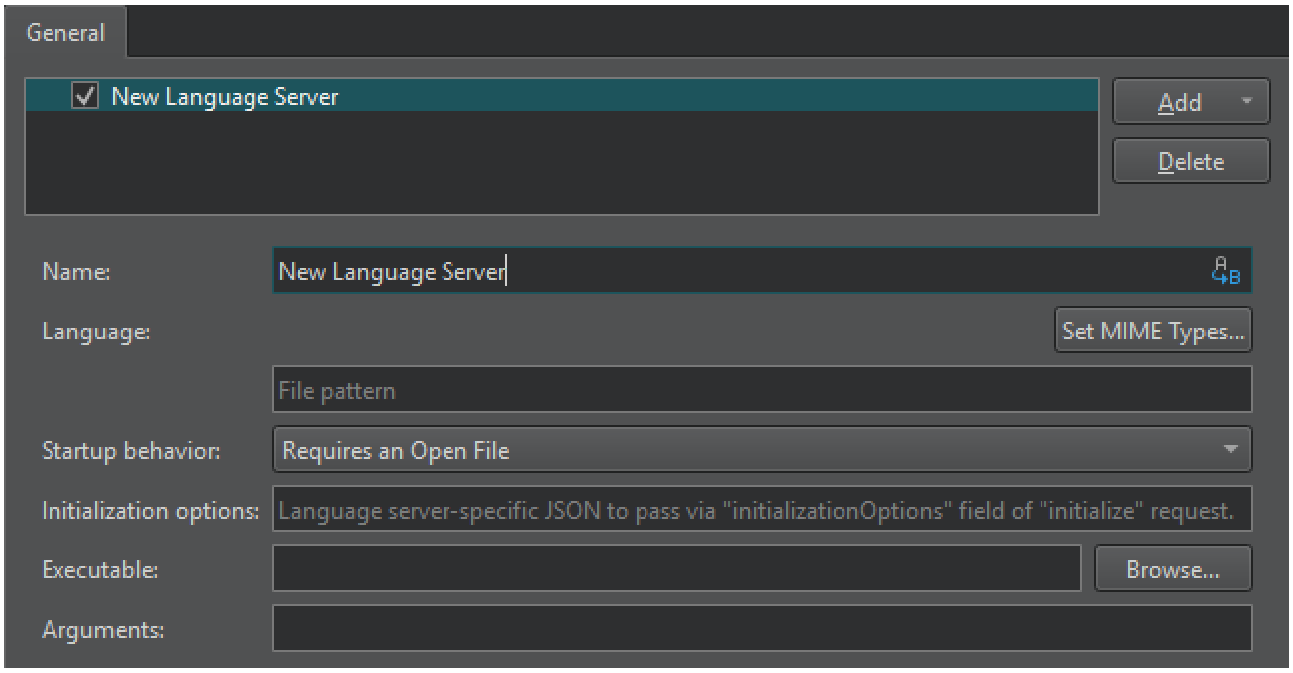
要添加语言服务器，请选择“**编辑**>**首选项**”>**语言客户端**>**添加**“（或在 macOS 上>**语言客户端**>**添加**”> **Qt Creator** >**首选项**“）。

要启用语言服务器，请选中语言服务器名称旁边的复选框并设置服务器首选项。

通用标准语言服务器

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。



New Language Server

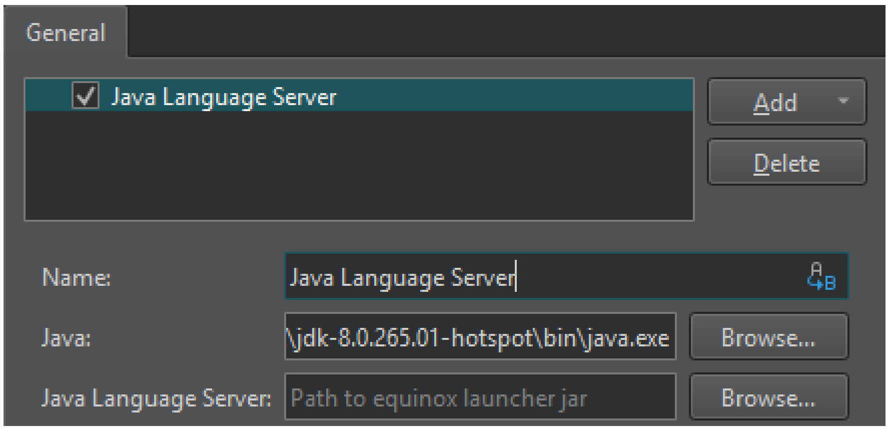
Name: New Language Server
Language: Set MIME Types...
File pattern
Startup behavior: Requires an Open File
Initialization options: Language server-specific JSON to pass via "initializationOptions" field of "initialize" request.
Executable: Browse...
Arguments:

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，以分号分隔。
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，可以添加特定于语言服务器的 JSON 属性以传递给请求。`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**”以将变量用作参数。

Java Language Server

To add a Java language server:

1. Select **Edit** > **Preferences** > **Language Client** > **Add** > **Java Language Server** to add a Java language server.



Java Language Server

Name: Java Language Server
Java: \jdk-8.0.265.01-hotspot\bin\java.exe Browse...
Java Language Server: Path to equinox launcher jar Browse...

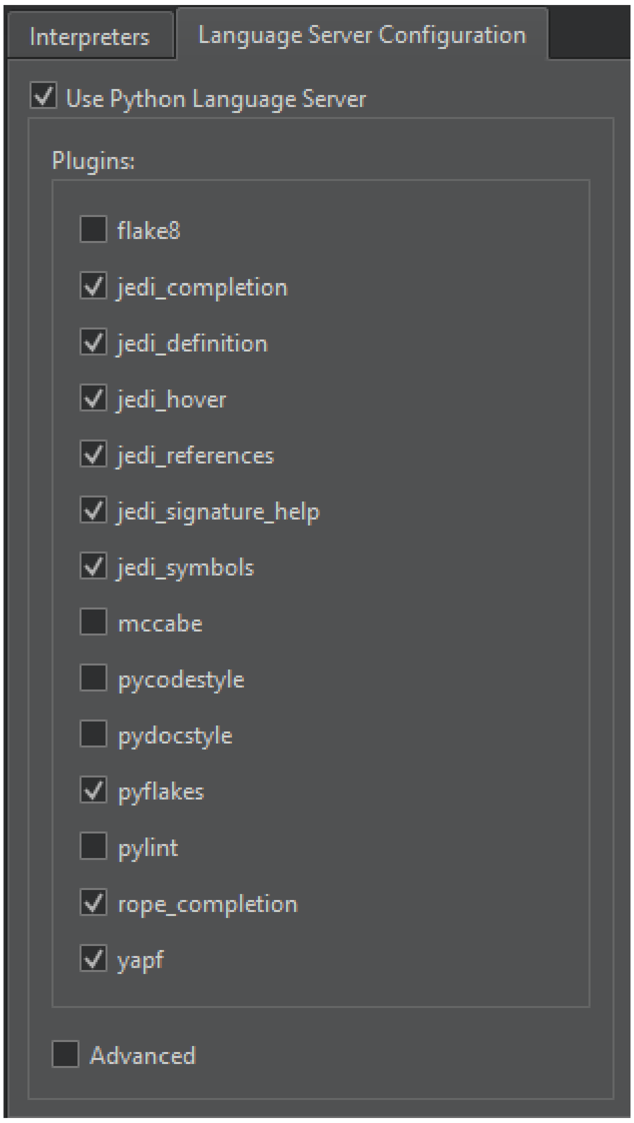
the server name. For more information, see Using Qt Creator Variables.

3. In the **Java** field, enter the path to the Java executable.
4. In the **Java Language Server** field, enter the path to the Java language server file. `.jar`

Python Language Server

To set preferences for Python language servers:

1. Select **Edit** > **Preferences** > **Python** > **Language Server Configuration** to select the Python language server plugins to use.



Interpreters | Language Server Configuration

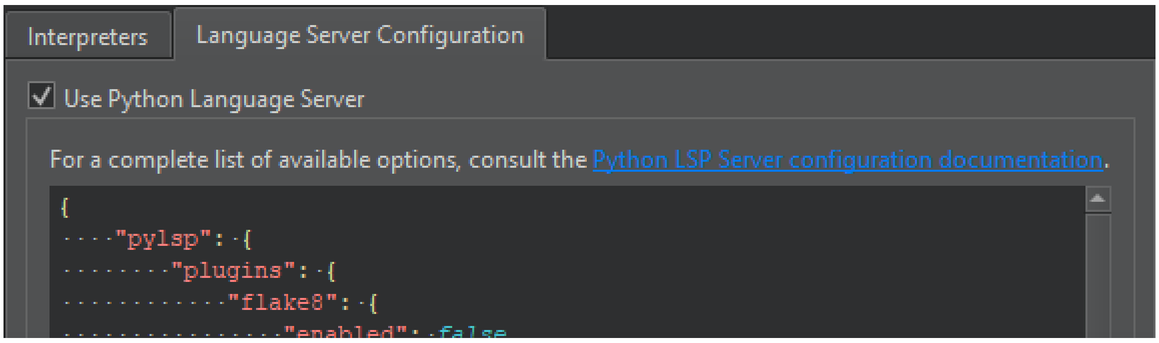
[✓] Use Python Language Server

Plugins:

- [ ] flake8
- [✓] jedi_completion
- [✓] jedi_definition
- [✓] jedi_hover
- [✓] jedi_references
- [✓] jedi_signature_help
- [✓] jedi_symbols
- [ ] mccabe
- [ ] pycodestyle
- [ ] pydocstyle
- [✓] pyflakes
- [ ] pylint
- [✓] rope_completion
- [✓] yapf

[ ] Advanced

2. Select **Advanced** to configure the plugins.

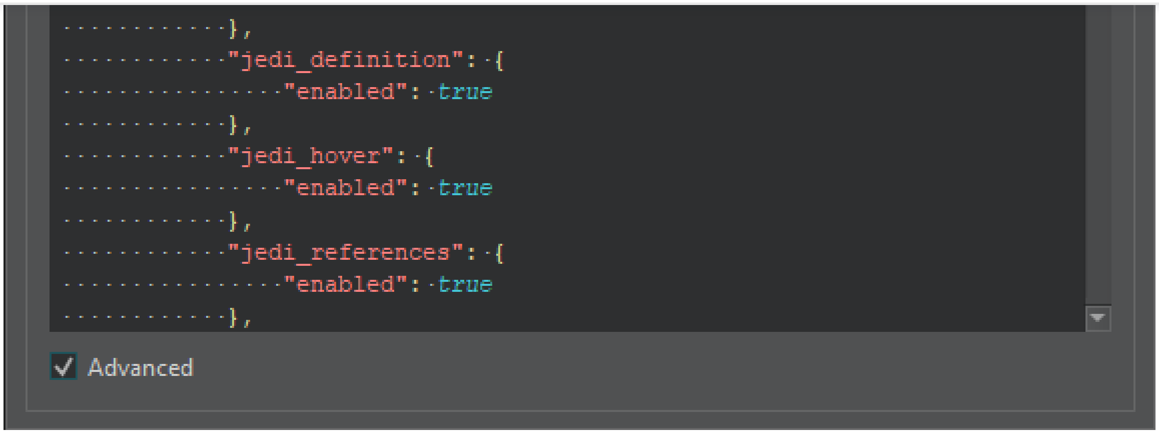


Interpreters | Language Server Configuration

[✓] Use Python Language Server

For a complete list of available options, consult the Python LSP Server configuration documentation.

```
{
    "pylsp": {
        "plugins": {
            "flake8": {
                "enabled": false
```



```
            },
            "jedi_definition": {
                "enabled": true
            },
            "jedi_hover": {
                "enabled": true
            },
            "jedi_references": {
                "enabled": true
            },
```

For a complete list of configuration options, see Python Language Server Configuration.

To disable the Python language server, deselect **Use Python Language Server**.

QML Language Server

Qt 6.4 ships with the language server that provides completion and warnings for QML. To set it up as a Generic StdIO Language Server, select and as MIME types, and as executable.`qmllstext/x-qmlapplication/x-qt.ui+qml<Qt Installation>/bin/qmlls`

If the language server is used together with the plugin, duplicate suggestions and warnings might be shown. To avoid this, disable the editor plugin as described in Enabling and Disabling Plugins.`QmlJSEditor`

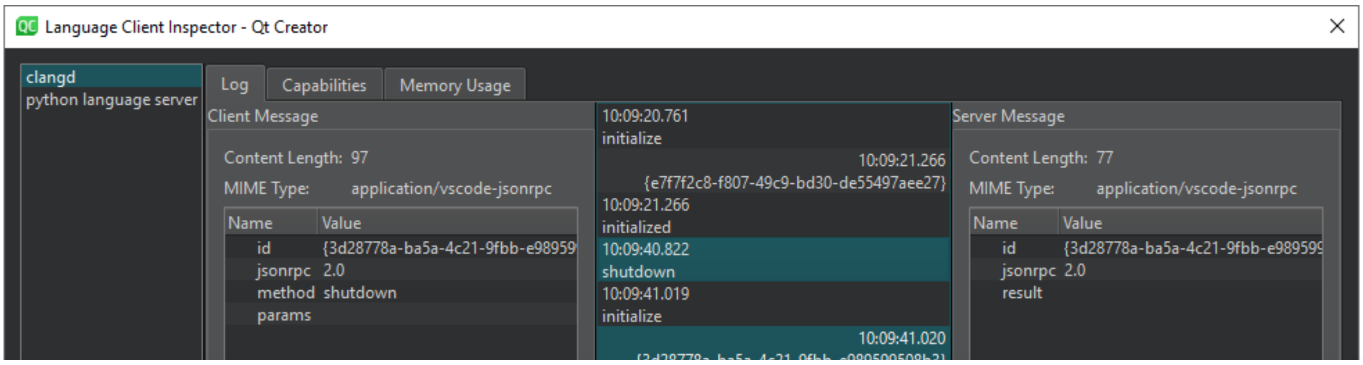
Supported Locator Filters

The locator enables you to browse not only files, but any items defined by *locator filters*. The language client plugin supports the following locator filters:

- Locating symbols in the current project (`:`)
- Locating symbols in the current document (`.`)
- Locating class (), enum, and function () definitions in your project`cm`

Inspecting Language Clients

Qt Creator sends messages (*Requests*) to the language server and receives responses that contain the requested information if the language server is capable of handling the requests. To inspect the communication between Qt Creator and language servers and view server capabilities, select **Tools** > **Debug Qt Creator** > **Inspect Language Clients**.



Client Message

Content Length: 97
MIME Type: application/vscode-jsonrpc

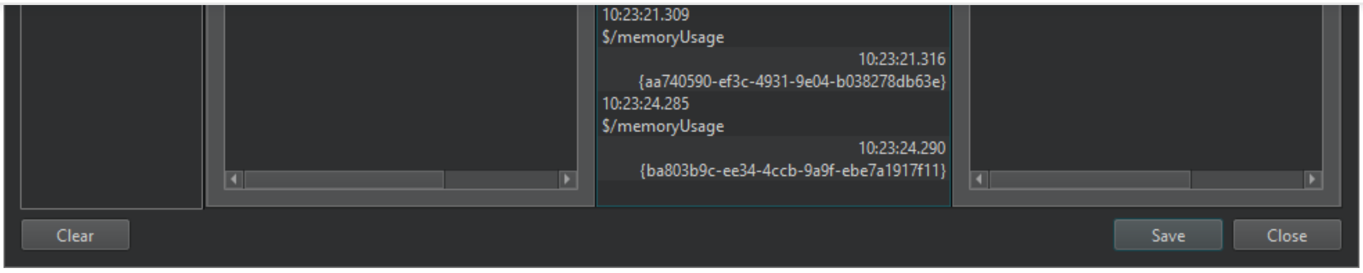
Name	Value
id	{3d28778a-ba5a-4c21-9fbb-e98959
jsonrpc	2.0
method	shutdown
params	

```
10:09:20.761
initialize
                                         10:09:21.266
         {e7f7f2c8-f807-49c9-bd30-de55497aee27}
10:09:21.266
initialized
10:09:40.822
shutdown
10:09:41.019
initialize
                                         10:09:41.020
```

Server Message

Content Length: 77
MIME Type: application/vscode-jsonrpc

Name	Value
id	{3d28778a-ba5a-4c21-9fbb-e989599
jsonrpc	2.0
result	


```
10:23:21.309
$/memoryUsage
                                  10:23:21.316
     {aa740590-ef3c-4931-9e04-b038278db63e}
10:23:24.285
$/memoryUsage
                                  10:23:24.290
     {ba803b9c-ee34-4ccb-9a9f-ebe7a1917f11}
```

The dialog shows a list of running language servers. The value of the **Startup behavior** field in the language server preferences determines when the server is started. The information displayed depends on the language server.

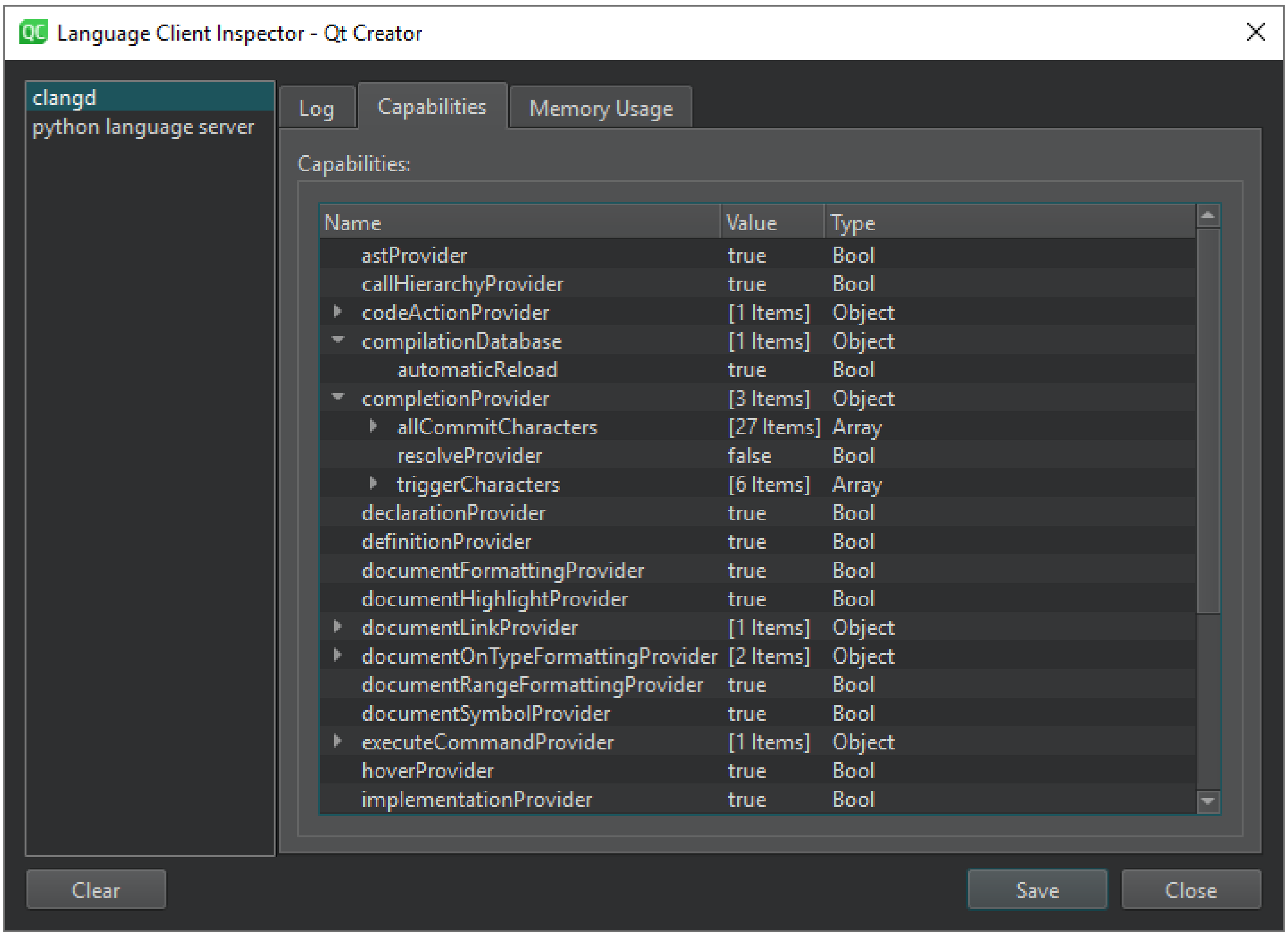
Log displays additional information about the selected log entry. You can see the **Content length** and **MIME type** of a **Client Message** and **Server Message**, as well as inspect the data sent between Qt Creator and the language server.

To remove old entries, select **Clear**.

Capabilities

In **Capabilities**, you can check whether a language server is capable of a specific task. You cannot modify the server capabilities in this dialog.

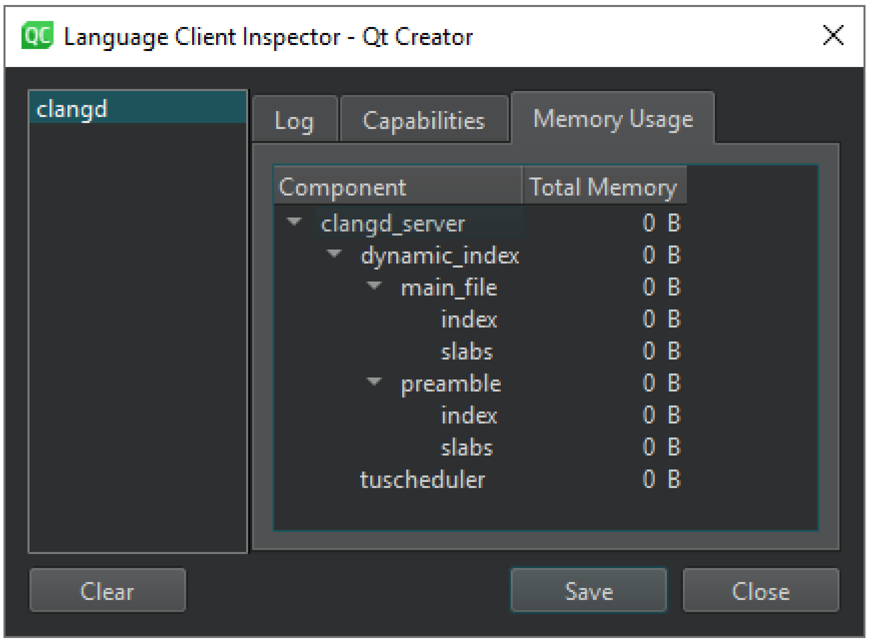
You can view the **Name**, **Value**, and **Type** of the capability.



Name	Value	Type
astProvider	true	Bool
callHierarchyProvider	true	Bool
codeActionProvider	[1 Items]	Object
compilationDatabase	[1 Items]	Object
automaticReload	true	Bool
completionProvider	[3 Items]	Object
allCommitCharacters	[27 Items]	Array
resolveProvider	false	Bool
triggerCharacters	[6 Items]	Array
declarationProvider	true	Bool
definitionProvider	true	Bool
documentFormattingProvider	true	Bool
documentHighlightProvider	true	Bool
documentLinkProvider	[1 Items]	Object
documentOnTypeFormattingProvider	[2 Items]	Object
documentRangeFormattingProvider	true	Bool
documentSymbolProvider	true	Bool
executeCommandProvider	[1 Items]	Object
hoverProvider	true	Bool
implementationProvider	true	Bool

For some language servers, **Dynamic Capabilities** lists the **Methods** and **Options** available.

Memory Usage



```
Component            Total Memory
▾ clangd_server              0 B
  ▾ dynamic_index            0 B
    ▾ main_file              0 B
        index                0 B
        slabs                0 B
    ▾ preamble               0 B
        index                0 B
        slabs                0 B
    tuscheduler              0 B
```

Reporting Issues

The language server client has been mostly tested with Python and Java. If problems arise when you try them or some other language, please select **Help** > **Report Bug** to report them in the Qt Project Bug Tracker. The reports should include Qt Creator console output with the environment variable set.`QT_LOGGING_RULES=qtc.languageclient.*=true`

‹ Using FakeVim Mode   Editing MIME Types ›

Qt DOCUMENTATION

Support

Support Services
Professional Services
Partners
Training

For Customers

Support Center
Downloads
Qt Login
Contact Us

Customer Success

Community

Contribute to Qt
Forum
Wiki
Downloads

Marketplace

Feedback Sign In