设置调试器

主调试器设置与生成和运行项目的工具包相关联。若要指定要用于每个工具包的调试器和编译器，请选择"**编辑>首选项>工具包**"。

仅当自动安装失败时，才需要设置调试器，因为缺少本机调试器（通常与 Windows 上的 CDB 调试器一样，您始终必须自己安装）或因为安装的版本不受支持（例如，当您的系统不包含或过时的 GDB 版本，并且您希望改用本地安装的替代项）。

> **注意：** 如果需要更改调试器以用于自动检测的工具包，则可以**克隆**工具包并更改克隆中的参数。确保为您的项目选择克隆的工具包。

如果未自动检测到要使用的调试器，请选择"**编辑>首选项**">**工具包>调试器>添加**"以添加调试器。

> **注意：** 若要使用 Windows 调试工具，必须安装这些工具，并将 Microsoft 提供的符号服务器添加到调试器的符号搜索路径中。有关更多信息，请参见在窗口上设置 CDB 路径。

> **注意：** 要在 macOS 上使用自由软件基金会 （FSF） 版本的 GDB，您必须对其进行签名并修改工具包设置。

本节介绍用于调试C++代码的选项，并提供受支持的本机调试器的安装说明。它也适用于其他编译语言的代码，如C，FORTRAN，Ada。

有关调试器模式的更多信息，请参见以不同模式启动调试器。

支持的本机调试器版本

Qt Creator 在处理已编译的代码时支持本机调试器。在大多数支持的平台上，可以使用 GNU 符号调试器 GDB。在微软视窗上，当使用微软工具链时，需要微软控制台调试器CDB。在 macOS 和 Linux 上，可以使用 LLDB 调试器。

下表总结了对调试C++代码的支持：

平台	编译器	本机调试器
操作系统	海湾合作委员会、国际商会	广发银行
独角蛋白	海湾合作委员会、国际商会	广发银行
苹果操作系统	海湾合作委员会，克朗	有限责任公司、自由软件基金会地理数据库（实验性）

视图/微软系统象像机	微软视觉C++编译器	适用于视图/CDB的调试工具

支持的地理数据库版本

从版本3.1开始，Qt创建者需要Python脚本扩展。不再支持没有Python脚本的GDB构建，并且将不起作用。支持的最低版本是使用Python版本2.7或3.3或更高版本的GDB 7.5。

对于使用 GDB 和 GDB 服务器的远程调试，目标设备上支持的最低 GDB 服务器版本为 7.0。

支持的中央数据库版本

Qt支持的所有版本的CDB目标平台都由Qt创建者支持。

支持的有限责任公司版本

LLDB 本机调试器具有与 GDB 调试器类似的功能。LLDB 是 macOS 上 Xcode 中的默认调试器，用于支持桌面上的C++。LLDB通常与Clang编译器一起使用（即使您也可以将其与GCC一起使用）。

在 macOS 上，您可以使用随 Xcode 一起提供的 LLDB 版本或从源代码构建。支持的最低版本是 LLDB 320.4。

在 Linux 上，支持的最低版本是 LLDB 3.8。

安装本机调试器

以下各节提供有关安装本机调试器的信息。

广发银行

在视窗上，使用支持 Python 的 GDB 版本，该版本与 Qt 包捆绑在一起，或者附带了最新版本的 MinGW。在大多数 Linux 发行版上，系统附带的 GDB 构建就足够了。

您还可以按照构建 GDB 中的说明构建自己的 GDB。

不再支持在 macOS 上随 Xcode 一起提供的 GDB 版本。

适用于窗口的调试工具

To use the CDB debugger, you must install the *Debugging tools for Windows*. You can download them from Download and Install Debugging Tools for Windows as part of the Windows SDK.

> **Note:** Visual Studio does not include the Debugging tools needed, and therefore, you must install them separately.

In addition, you must select **Qt Creator CDB Debugger Support** (in **Qt** > **Tools** > **Qt Creator**) when you install Qt or the stand-alone Qt Creator.

When manually building Qt Creator using the Microsoft Visual C++ Compiler, the build process checks for the required files in `."%ProgramFiles%\Debugging Tools for Windows"`

It is highly recommended that you add the Symbol Server provided by Microsoft to the symbol search path of the debugger. The Symbol Server provides you with debugging informaton for the operating system libraries for debugging Windows applications. For more information, see Setting CDB Paths on Windows.

The Qt binary distribution contains both debug and release variants of the libraries. But you have to explicitly tell the runtime linker that you want to use the debug libraries even if your application is compiled as debug, as release is the default library.

If you use a qmake based project in Qt Creator, you can set a flag in your run configuration, in **Projects** mode. In the run configuration, select **Use debug version of frameworks**.

For more detailed information about debugging on macOS, see: Mac OS X Debugging Magic.

LLDB

We recommend using the LLDB version that is delivered with the latest Xcode.

Setting up FSF GDB for macOS

To use FSF GDB on macOS, you must sign it and add it to the Qt Creator kits.

1. To create a key for signing FSF GDB, select **Keychain Access** > **Certificate Assistant** > **Create a Certificate**:
    1. In the **Name** field, input **fsfgdb** to replace the existing content.
    2. In the **Certificate Type** field, select **Code Signing**.
    3. Select the **Let me override defaults** check box.
    4. Select **Continue**, and follow the instructions of the wizard (use the default settings), until the **Specify a Location For The Certificate** dialog opens.
    5. In the **Keychain** field, select **System**.
    6. Select **Keychain Access** > **System**, and locate the certificate.
    7. Double click the certificate to view certificate information.
    8. In the **Trust** section, select **Always Trust** in the **When using this certificate** field, and then close the dialog.
2. To sign the binary, enter the following command in the terminal:

```
codesign -f -s "fsfgdb" $INSTALL_LOCATION/fsfgdb
```

3. In Qt Creator, select **Qt Creator** > **Preferences** > **Kits** > **Add** to create a kit that uses FSF GDB.
4. In the **Debugger** field, specify the path to FSF GDB (, but with an explicit value for).`$HOME/gdb72/bin/fsfgdb$HOME`
5. To use the debugger, add the kit in the **Build Settings** of the project.

‹ Debugging

Launching the Debugger ›

Qt DOCUMENTATION

Qt Company

Contact Us

Company

About Us
Investors
Newsroom
Careers
Office Locations

Licensing

Terms & Conditions
Open Source
FAQ

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