

[Qt创作者手册](#) > [连接安卓设备](#)

连接安卓设备

您可以使用USB电缆将Android设备连接到开发PC，以从Qt Creator构建，运行，调试和分析应用程序。使用 Qt 5 进行开发时，支持 Android 版本 4.1（API 级别 16）或更高版本的设备，使用 Qt 6 进行开发时，支持 Android 版本 6.0（API 级别 23）的设备。

若要针对 Android 进行开发，必须具有用于在开发电脑上安装的 Android 设备构建应用程序的工具链。Qt Creator可以自动下载和安装工具链，并创建一个合适的构建和运行**工具包**，其中包含工具链和适用于设备架构的AndroidQt版本。

从Qt 5.14.0开始，Qt for Android软件包包含所有安装的架构（ABI）。

要为 Java 启用有用的代码编辑功能（如代码完成、突出显示、函数工具提示和在代码中导航），请添加**Java 语言服务器**。

Android Debug Bridge（adb）命令行工具集成到Qt Creator中，使您能够将应用程序部署到连接的Android设备，运行它们并读取其日志。它包括在开发主机上运行的客户端和服务端，以及在模拟器或设备上运行的守护程序。

要求

要使用Qt Creator开发适用于Android的Qt应用程序，您需要Qt for Android 5.2或更高版本，以及Qt Creator可以自动为您下载、安装和配置的工具链。有关详细信息，请参阅**手动安装必备组件**。

指定安卓设备设置

Qt Creator提供自动安装所有必要的软件包和工具，并通过创建调试器，工具链和工具包来设置您的开发环境。您可以使用Qt Creator执行以下操作：

- › 下载并解压缩 Android SDK 命令行工具。
- › 安装或更新基本软件包，例如 NDK、构建工具和平台工具。

要设置适用于 Android 的开发环境，请执行以下操作：

1. 在 Windows 和 Linux 上选择**Edit>Preferences>Devices>Android** 或在 macOS 上选择**Qt Creator>Preferences>Devices>Android**。

Devices

Android

WebAssembly

QNX

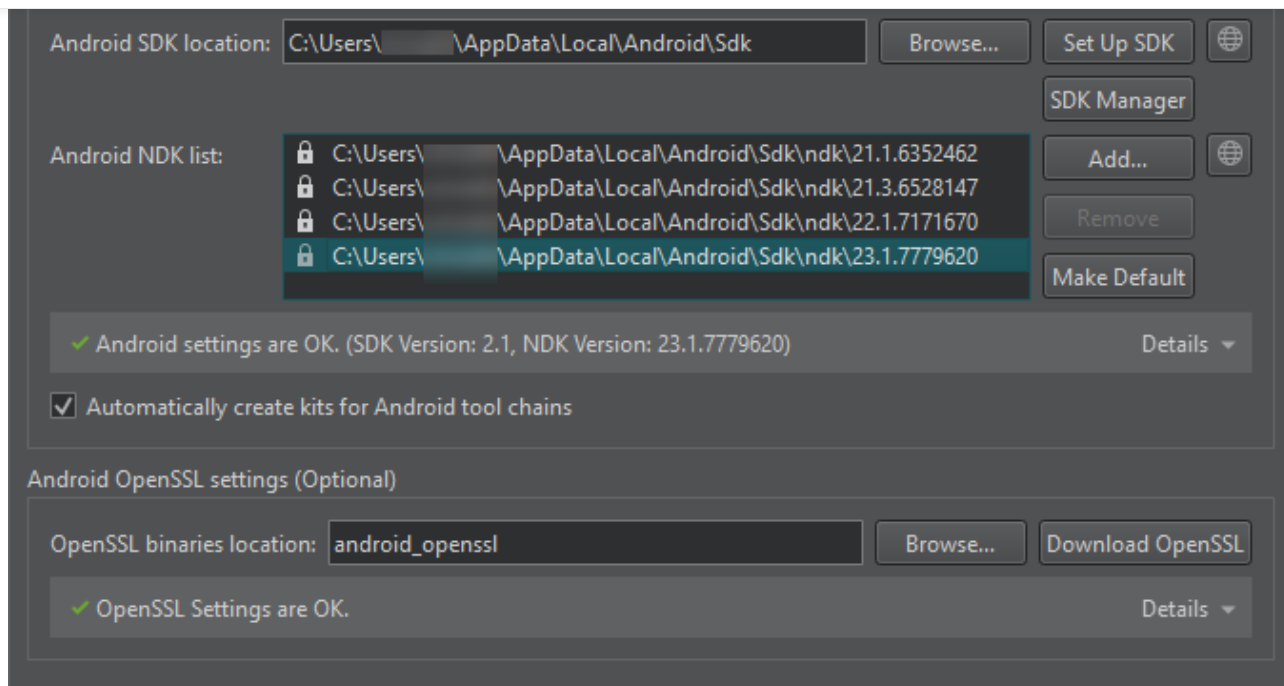
Bare Metal

SSH



DOCUMENTATION





2. 在 **JDK 位置** 字段中，设置 JDK 的路径。Qt Creator 会检查 JDK 安装并报告错误。

默认情况下，Qt Creator 会尝试查找受支持的 **AdoptOpenJDK** 或 **OpenJDK** 安装。如果未找到路径，则必须手动设置路径。如果未安装受支持的 JDK，请选择在 默认浏览器中打开 JDK 下载网页。

注意：我们建议使用 64 位 JDK，因为 32 位 JDK 可能会导致问题，并且某些包可能未列出。
cmdline-tools

3. 在 Android SDK **位置** 字段中，设置要安装 **Android SDK 命令行工具** 的文件夹的路径。

4. 选择“设置 SDK”以自动下载 Android SDK 命令行工具并将其提取到所选路径。

SDK 管理器会检查工具链是否已安装。如果缺少包或需要更新，SDK 管理器会提供添加或删除这些包。在执行操作之前，它会提示您接受即将进行的更改。此外，如有必要，它会提示您接受 Google 许可。

5. 已安装的 NDK 版本列在 **安卓 NDK 列表中**。锁定的项目由 SDK 管理器安装，只能从 **Android SDK 管理器** 对话框中进行修改。有关更多信息，请参阅 **管理安卓 NDK 软件包**。
6. 选中“**自动为 Android 工具链创建工具包**”复选框，以允许 Qt Creator 为您创建工具包。如果找不到合适的 Qt 版本，Qt Creator 会显示警告。
7. （可选）在 Android OpenSSL 设置组中，设置预构建的 **OpenSSL** 库的路径。

对于需要 OpenSSL 支持的 Qt 应用程序，Qt Creator 允许快速将 **Android OpenSSL 支持** 添加到您的项目中。有关更多信息，请参见 **添加外部库**。

8. 选择“**下载 OpenSSL**”，将 OpenSSL 存储库下载到所选路径。如果自动下载失败，下载网页将打开以供手动下载。

手动设置

注意：我们建议您使用最新的安卓 SDK 命令行工具。不支持使用 Android SDK Tools 版本 25.2.5 或更早版本，因为它们无法与 Qt Creator 完全集成。

但是，如果您无法设置或无法进口您的系统，您可以下载并安装 Android Studio 的 SDK 平台工具，然后安装或更新开发所

查看安卓工具链设置

Android SDK 命令行工具下载 URL、基本软件包列表以及每个 Qt 版本的相应 NDK 在 JSON 配置文件中定义。该文件位于用户的 Qt Creator 资源文件夹下：

```
# Linux and macOS
~/.config/QtProject/qtcreator/android/sdk_definitions.json

# Windows
C:\Users\Username\AppData\Local\QtProject\qtcreator\android\sdk_definitions.json
```

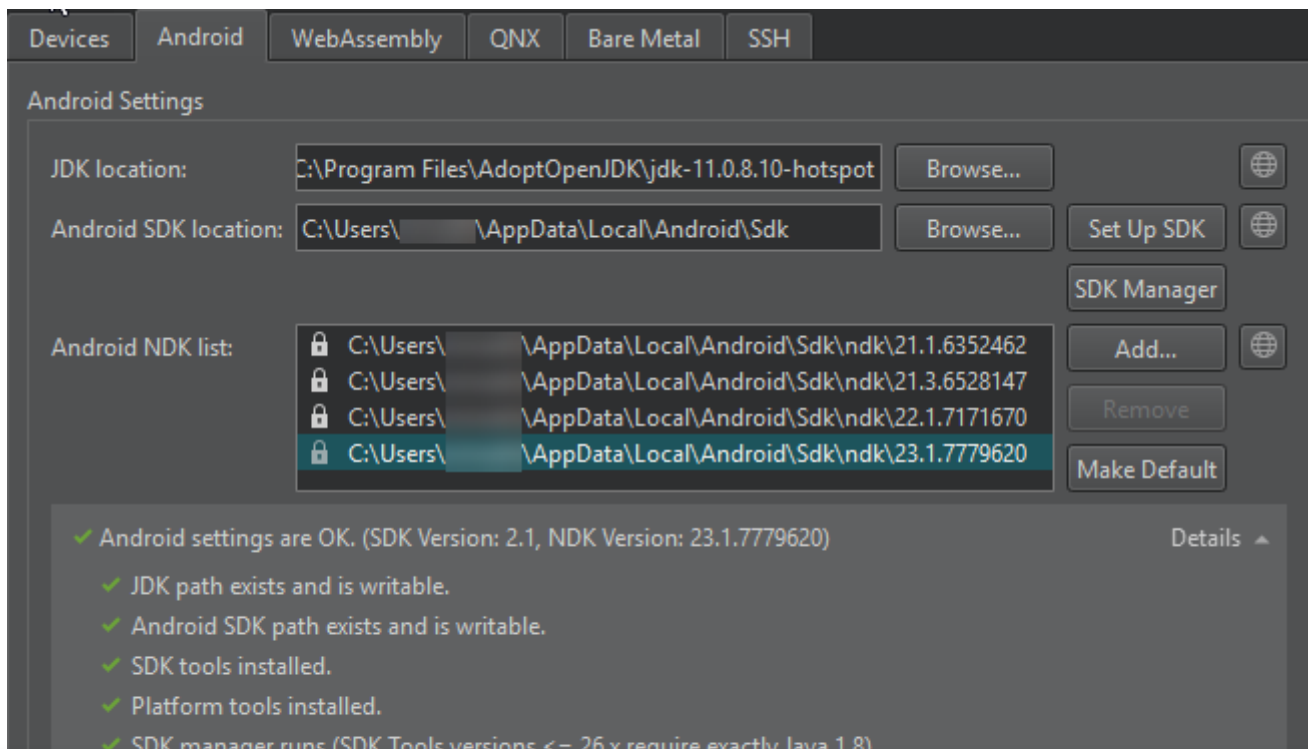
例如，SDK 配置文件定义了要用于 Qt 5.12.0 至 5.12.5 和 Qt 5.13.0 至 5.13.1 版本的 NDK 版本 19.2.5345600：

```
"specific_qt_versions": [
  {
    "versions": ["5.12.[0-5]", "5.13.[0-1]"],
    "sdk_essential_packages": ["build-tools;28.0.2", "ndk;19.2.5345600"],
    "ndk_path": "ndk/19.2.5345600"
  }
]
```

您可以在 Git 中查看最新版本的配置文件，其中包含 Android SDK 和 NDK 更改 [sdk_definitions.json](#)。

管理安卓 NDK 软件包

要查看已安装的 Android NDK 版本，请在 Windows 和 Linux 上选择 **Edit > Preferences > Devices > Android** 或在 macOS 上选择 **Qt Creator > Preferences > Devices > Android**。



✓ All essential packages installed for all installed Qt versions.

锁定的版本由 SDK 管理器安装，只能从Android SDK 管理器对话框中进行修改。有关更多信息，请参阅[管理 Android SDK 包](#)。

要手动下载 NDK，请选择。

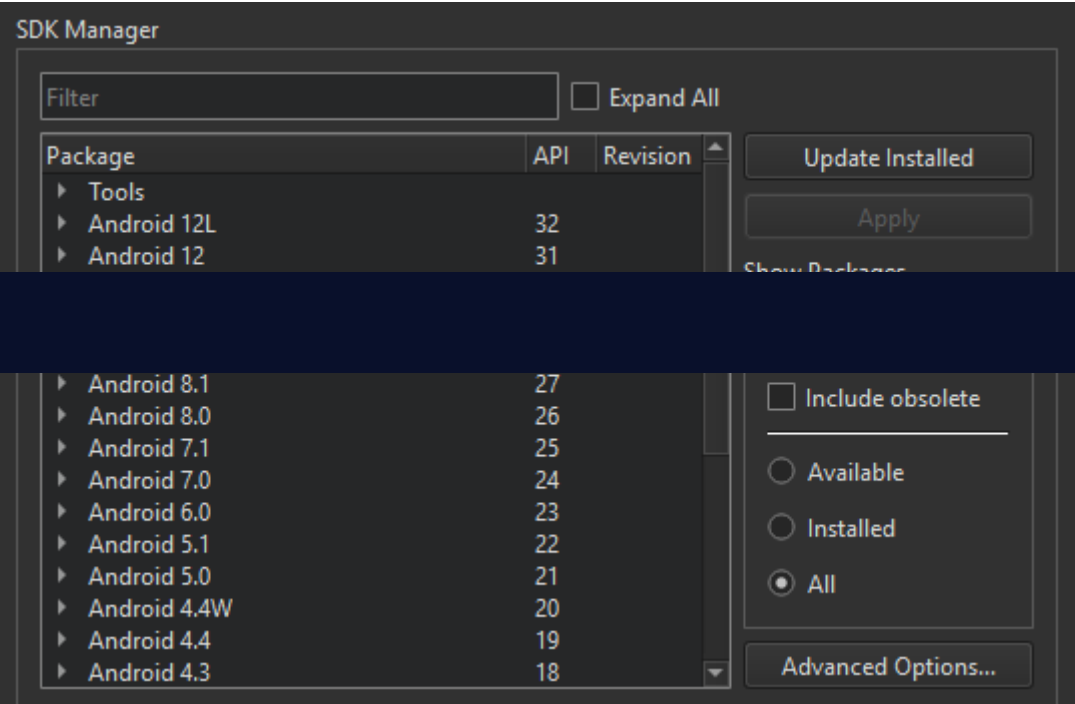
要默认对所有Qt版本使用选定的NDK版本，请选择**设为默认版本**。

要手动将自定义 NDK 路径添加到全局 NDK 列表，请选择**添加**。这将创建与该 NDK 关联的自定义工具链和调试器。但是，您必须手动创建使用自定义 NDK 的工具包。有关详细信息，请参阅[添加工具包](#)。

管理安卓开发工具包

从 Android SDK Tools 版本 25.3.0 开始，Android 仅提供命令行工具 `sdkmanager` 用于 SDK 包管理。为了使SDK管理更容易，Qt Creator提供了一个SDK管理器，用于安装、更新和删除SDK包。您仍然可以使用 `sdkmanager` 进行高级 SDK 管理。

要查看已安装的 Android SDK 软件包，请在 Windows 和 Linux 上选择Edit>Preferences>Devices>Android>SDK Manager，或在 macOS 上选择Qt Creator>Preferences>Devices>Android>SDK Manager。



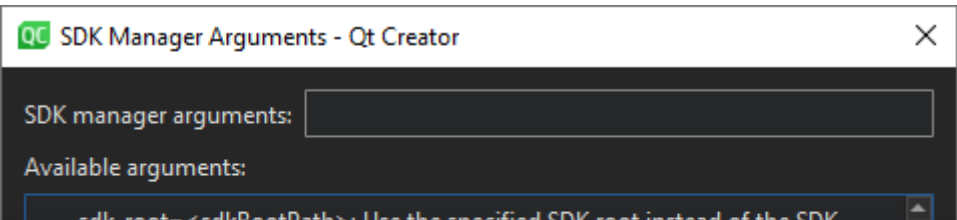
Qt创建者手册8.0.2

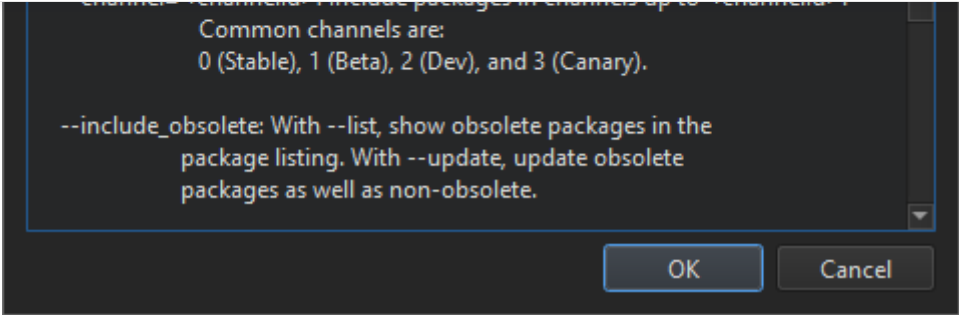
Topics >

可以显示你在“显示包>频道”中选择的发布频道的包。常见的频道 ID 包括**稳定**、**测试**、**开发**和**金丝雀**。若要显示和更新过时的包，请选择“**包括过时的包**”。若要筛选包，请选择“**可用**”、“**已安装**”或“**全部**”。

要更新已安装的 Android SDK 软件包，请选择**更新已安装**。选择要更新的程序包，然后选择“**应用**”。

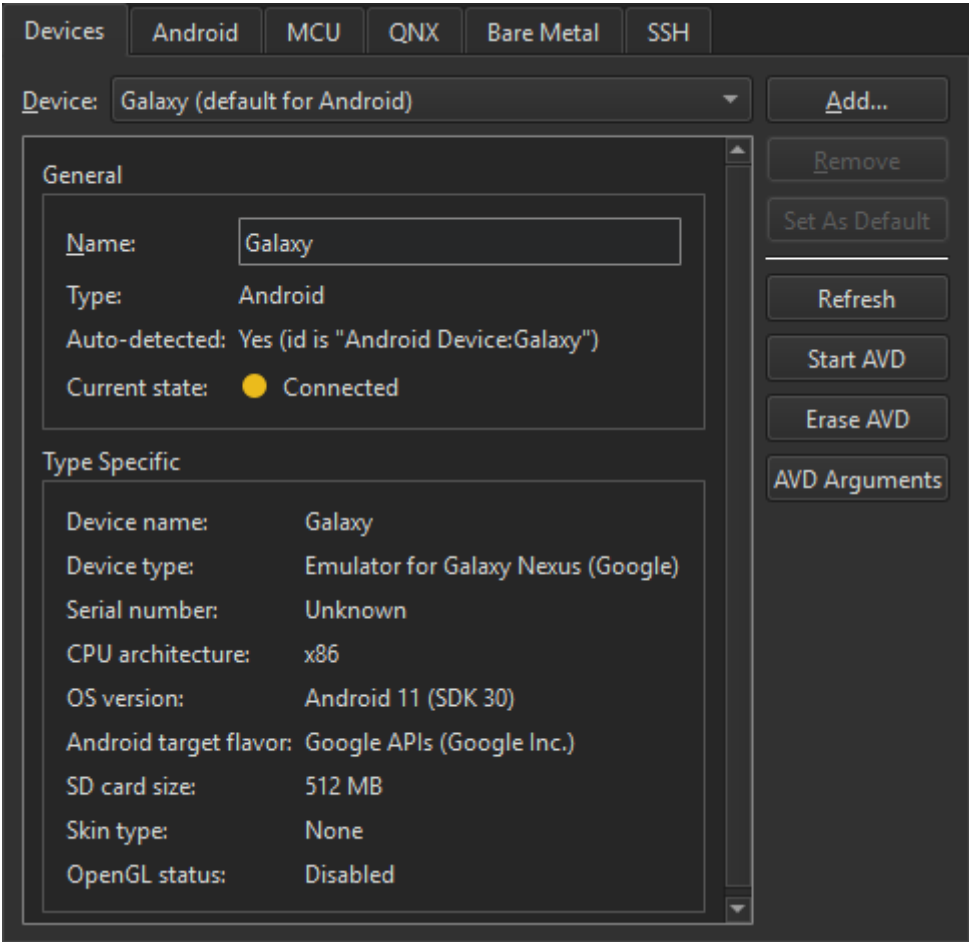
要指定高级 `sdkmanager` 设置，请选择**高级选项**，然后在SDK 管理器参数字段中输入参数。可用参数中列出并描述了可用**参数**。





管理安卓虚拟设备（AVD）

可用的 AVD 列在 Windows 和 Linux 上的Edit>Preferences>Devices 或 macOS 上的Qt Creator>Preferences>Devices> 中。您可以添加更多 AVD。

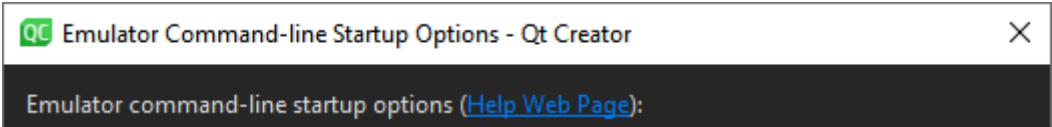


您可以在“当前状态”中查看所选设备的状态。若要更新状态信息，请选择“刷新”。

要启动AVD，请选择**启动 AVD**。通常，您不需要单独启动 AVD，因为当您在**套件选择器**中选择它们以**将应用程序部署到**它们时，它们会自动启动。

要从列表和套件选择器中删除 AVD，请选择**擦除 AVD**。

要指定用于启动 AVD 的选项，请选择**AVD 参数**。



在**模拟器命令行启动**选项中指定选项。有关可用选项，请参阅[从命令行启动模拟器](#)。

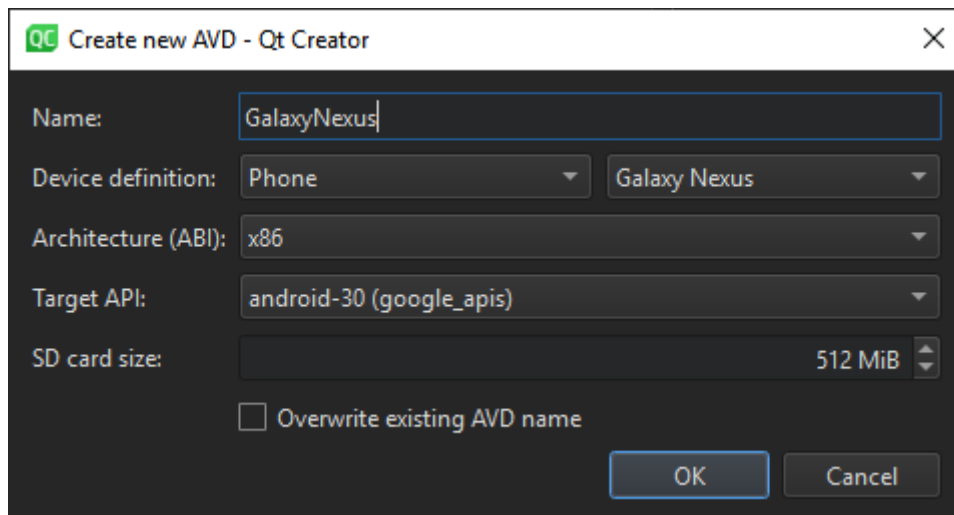
注意：Android 模拟器有一个错误，阻止它在某些系统上启动。如果 AVD 未启动，您可以尝试通过运行以下命令手动启动它：

```
cd <ANDROID_SDK>/emulator
./emulator -avd <AVD_NAME>
```

创建新的 AVD

要创建新的虚拟设备：

1. 在 Windows 和 Linux 上选择 **Edit>Preferences>Devices>Add>Android Device** 或 **macOS 上的 Qt Creator>Preferences>Devices>Add>Android 设备** 以打开“**创建新的 AVD**”对话框。



2. 设置设备的名称、定义、体系结构、目标 API 级别和 SD 卡大小。
3. 选择**确定**以创建 AVD。

有关创建新 AVD 的更多高级选项，请使用命令行工具 [avdmanager](#) 或 Android Studio 的原生 AVD 管理器界面。

在安卓设备上调试

调试在不同的 Android 设备上以不同的方式启用。在“**开发人员选项**”下查找**USB调试**。在某些设备上，“**开发人员选项**”是隐藏的，并且仅在您多次点击“**设置>关于**”中的**内部版本号**字段时才可见。有关详细信息，请参阅[配置设备上的开发人员选项](#)。

选择调试生成[配置以生成用于调试](#)的应用程序。

注意：如果 Android Studio 正在运行，Qt Creator 无法调试 Android 设备上的应用程序。如果[应用程序输出](#)中显示以下消息，请关闭 Android 工作室并重试：

[连接设备](#)

[连接裸机设备](#)

©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

