

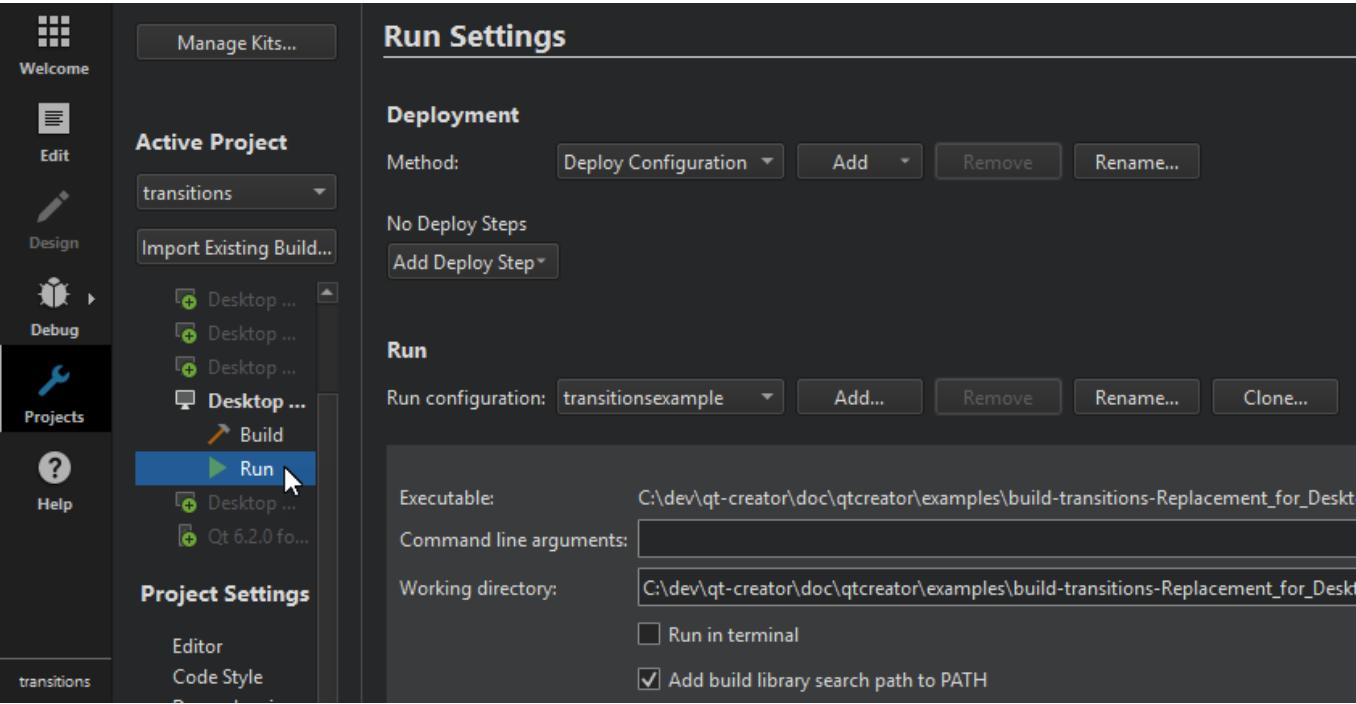
🔍 搜索

Qt创作者手册 > [指定运行设置](#)

指定运行设置

要指定的运行设置取决于项目的类型以及生成和运行项目时使用的[工具包](#)。

Qt Creator会自动为您的项目创建运行配置。若要查看和修改它们，请选择“**项目>生成和运行>运行**”。



要防止Qt Creator自动创建运行配置，请选择Edit>Preferences>Build & Run，然后取消选中“**自动创建合适的运行配置**”复选框。

管理运行配置

可用的运行配置列在**运行配置**字段中。若要为项目添加运行配置，请选择“**添加**”。若要添加基于当前配置的运行配置，请选择“**克隆**”。

若要重命名当前运行配置，请选择“**重命名**”。

若要删除当前运行配置，请选择“**删除**”。

qmake 项目的运行配置从解析的 .pro 文件中派生其可执行文件。有关如何构造命令的详细信息，请参阅[启动外部进程](#)。

如果一个项目有多个可执行文件，您需要告诉Qt Creator它应该运行哪一个。

CMake 运行目标

使用 CMake 时，可以通过在项目中将属性的值设置为来筛选运行目标列表。例如：
qtc_runnableFOLDERCMakeLists.txt

```
set_target_properties(main_executable PROPERTIES FOLDER "qtc_runnable")
```

如果您没有为任何项目指定，Qt Creator会自动为中指定的所有目标添加运行配置。
qtc_runnableCMakeLists.txt

qmake 运行目标

使用 qmake 时，您可以通过在要运行的应用程序项目（）的 .pro 文件中指定变量来防止 Qt Creator 自动为子项目创建运行配置。例如qtc_runnableTEMPLATE=app

```
CONFIG += qtc_runnable
```

如果您没有指定任何应用程序项目，Qt Creator会为所有应用程序项目创建运行配置。qtc_runnable

如果您的任何应用程序项目指定，Qt Creator 仅为在其 .pro 文件中设置的子项目创建运行配置。
qtc_runnableCONFIG += qtc_runnable

有关 qmake 项目模板的详细信息，请参阅[模板](#)。

介子运行目标

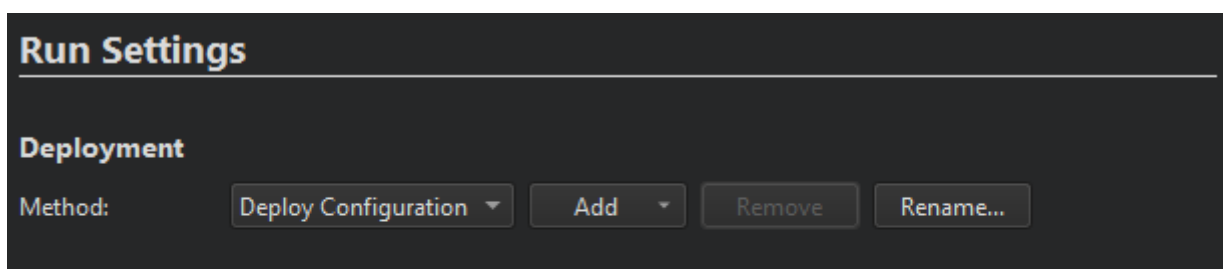
Qt Creator会自动为Meson构建描述中声明的所有目标添加运行配置。executable()

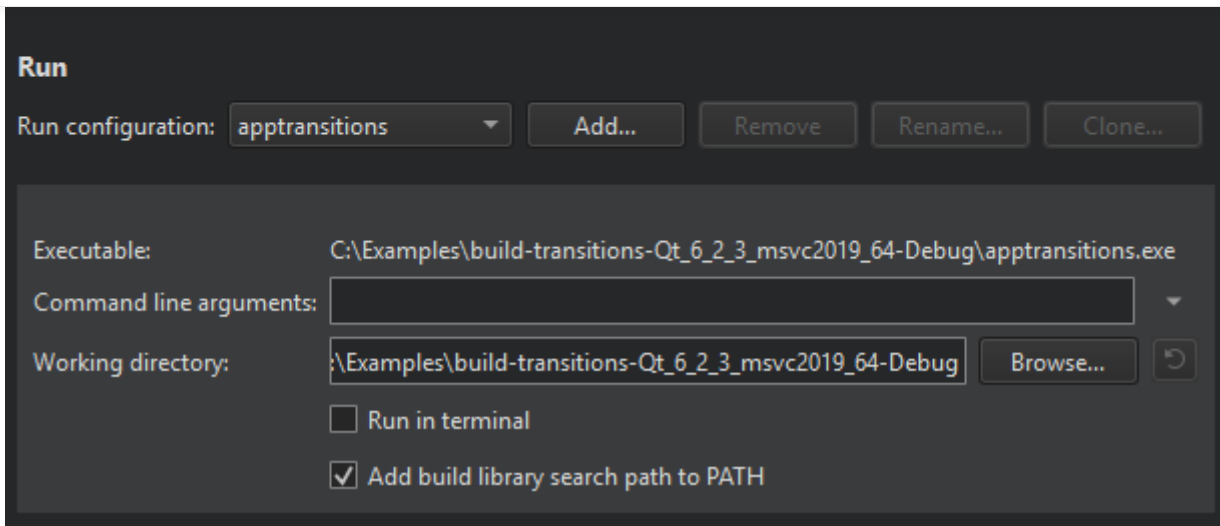
指定桌面设备类型的运行设置

您可以指定要传递给可执行文件的命令行参数以及要使用的工作目录。工作目录默认为生成结果的目录。

对于控制台应用程序，请选中在**终端中运行**复选框。要指定要在 Linux 和 macOS 上使用的终端，请选择**编辑>首选项>环境>系统**。

若要使用设置的特殊环境变量运行，请在“**运行环境**”部分中选择它们。有关更多信息，请参阅[选择运行环境](#)。





构建应用程序时，Qt Creator会创建一个目录列表，链接器将在其中查找应用程序链接到的库。默认情况下，链接库对Qt Creator尝试运行的可执行文件可见。通常，仅当此选项会导致不必要的副作用或使用部署步骤（例如）并希望确保部署的应用程序在没有Qt Creator的情况下运行时也能找到库时，才应禁用此选项。make install

要禁用当前项目的库链接，请取消选中“**将构建库搜索路径添加到 PATH**”复选框。若要禁用所有项目的库链接，请选择“**编辑>首选项>生成和运行**”，然后取消选中“**将链接器库搜索路径添加到运行环境**”复选框。

使用框架的调试版本（DYLD_IMAGE_SUFFIX=_debug）选项（仅在 macOS 上可用）使您能够调试（例如，单步执行）链接框架，例如 Qt 框架本身。调试应用程序代码不需要此选项。

在 Linux 上，选中“以根用户身份运行”复选框以使用**根用户权限运行**应用程序。

您还可以创建自定义可执行运行配置，在其中可以设置要运行的可执行文件。有关更多信息，请参见[指定要运行的自定义可执行文件](#)。

指定瓦尔格林德设置

Qt Creator集成了[Valgrind代码分析工具](#)，用于检测内存泄漏和分析函数执行。您可以根据需要配置工具。

您可以为所有项目全局指定 Valgrind 设置，也可以为每个项目单独指定 Valgrind 设置。

要为当前项目指定瓦尔格林德设置，请执行以下操作：

1. 在“**瓦尔格林德设置**”部分中，选择“**自定义**”。
2. 指定项目的瓦尔格林德设置。



3. 在 Valgrind 可执行文件中，指定Valgrind **可执行文件**的路径。

除文件后端以外之外的所有位置。

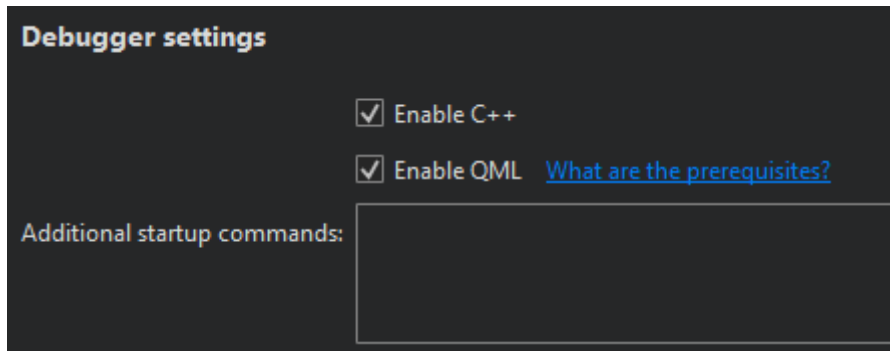
有关 CallGrind 和 MemCheck 设置的详细信息，请参阅：

- › [选择性能分析选项](#)
- › [选择内存分析选项](#)

单击“恢复全局”以恢复为**全局**设置。

要指定全局 Valgrind 设置，请选择**编辑>首选项>分析器**。

启用调试



若要选择要调试的语言，请选中“**启用C++**”和“**启用 QML**”复选框。

注意：在已知端口打开套接字会带来安全风险。互联网上的任何人都可以连接到您正在调试的应用程序并执行任何JavaScript函数。因此，必须确保端口受到防火墙的正确保护。

(可选) 在“**其他启动命令**”中，可以输入用于调试C++的其他设置：

- › [自定义调试帮助程序](#)
- › 在 GDB 启动之后、启动或附加调试程序之前以及初始化调试帮助程序之前执行的 [GDB 命令](#)

但是，您通常可以将此字段留空。

有关调试的详细信息，请参阅[调试](#)。

指定安卓设备的运行设置

要在 Android 设备上运行和调试应用程序，您必须按照连接[Android 设备](#)中的说明创建从开发主机到设备的连接。

启动应用程序时，将应用一组默认的 Android 活动管理器（am）启动选项。您可以在【**作业管理器**】**开始参数**字段中指定其他开始选项。但是，如果默认选项与添加的选项冲突，则应用程序可能无法启动。

正常运行的活动管理器的默认参数：

```
am start -n <package_name>/<QtActivity_name>
```

```
am start -n <package_name>/<QtActivity_name> -D
```

例如，要以特定用户身份运行应用程序，请输入启动选项，其中用户帐户的用户 ID。--user 1010

The screenshot shows the 'Run' configuration window for a configuration named 'cmake_example'. It includes buttons for 'Add...', 'Remove', 'Rename...', and 'Clone...'. The 'Command line arguments' field is empty. The 'Activity manager start arguments' field contains '--user 10'. The 'Pre-launch on-device shell commands' field contains 'am switch-user 10' and 'input keyevent 82'. The 'Post-quit on-device shell commands' field contains 'am switch-user 0' and 'input keyevent 82'.

您可以指定要在应用程序启动之前和退出之后运行的 shell 命令。例如，在**预启动设备上的 shell 命令**中输入以下命令，以解锁屏幕并在运行应用程序之前切换到设备上的用户帐户：10

```
input keyevent 82
am switch-user 10
```

在**退出后设备上的 shell 命令**中输入以下命令，以切换回默认用户，并在退出应用程序后解锁屏幕：0

```
am switch-user 0
input keyevent 82
```

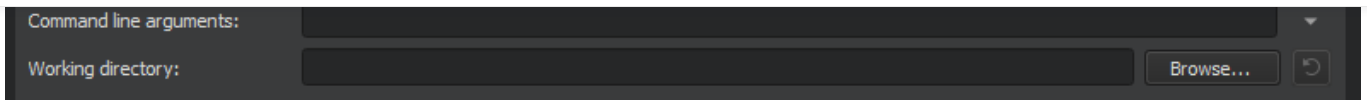
为基于 Linux 的设备指定运行设置

若要在基于 Linux 的设备上运行和调试应用程序，必须创建从开发主机到设备的连接，并将设备配置添加到**工具包**。单击**管理套件**以将设备添加到套件。有关详细信息，请参阅[连接通用远程 Linux 设备](#)。

当您运行应用程序时，Qt Creator 会将文件复制到连接的设备。

运行设置显示开发主机和设备上可执行文件的路径。例如，若要指定其他应用程序启动您的应用程序，请在“**设备上的备用可执行文件**”字段中输入命令，然后选中“**改用此命令**”复选框。

The screenshot shows the 'Run' configuration window for a configuration named 'photos (on Generic Linux Device)'. It includes buttons for 'Add...', 'Remove', 'Rename...', and 'Clone...'. The 'Executable on device' field shows the path '/opt/photos/bin/photos'.



您可以在命令行参数字段中指定要传递给应用程序的**参数**。

指定 QNX 设备的运行设置

要在 QNX 设备上运行和调试应用程序，您必须创建从开发 PC 到设备的连接。单击**管理设备配置**以创建连接。有关更多信息，请参阅[连接 QNX 设备](#)。

为 QNX Neutrino 设备指定**运行设置**与为 **Linux 设备指定运行设置**非常相似。

指定 Boot2Qt 设备的运行设置

要在 **Boot2Qt** 设备上运行和调试应用程序（仅限商业），必须创建从开发主机到设备的连接，并将设备配置添加到**工具包**。选择“**管理工具包**”以将设备添加到工具包。有关更多信息，请参阅[Boot2Qt：安装指南](#)。

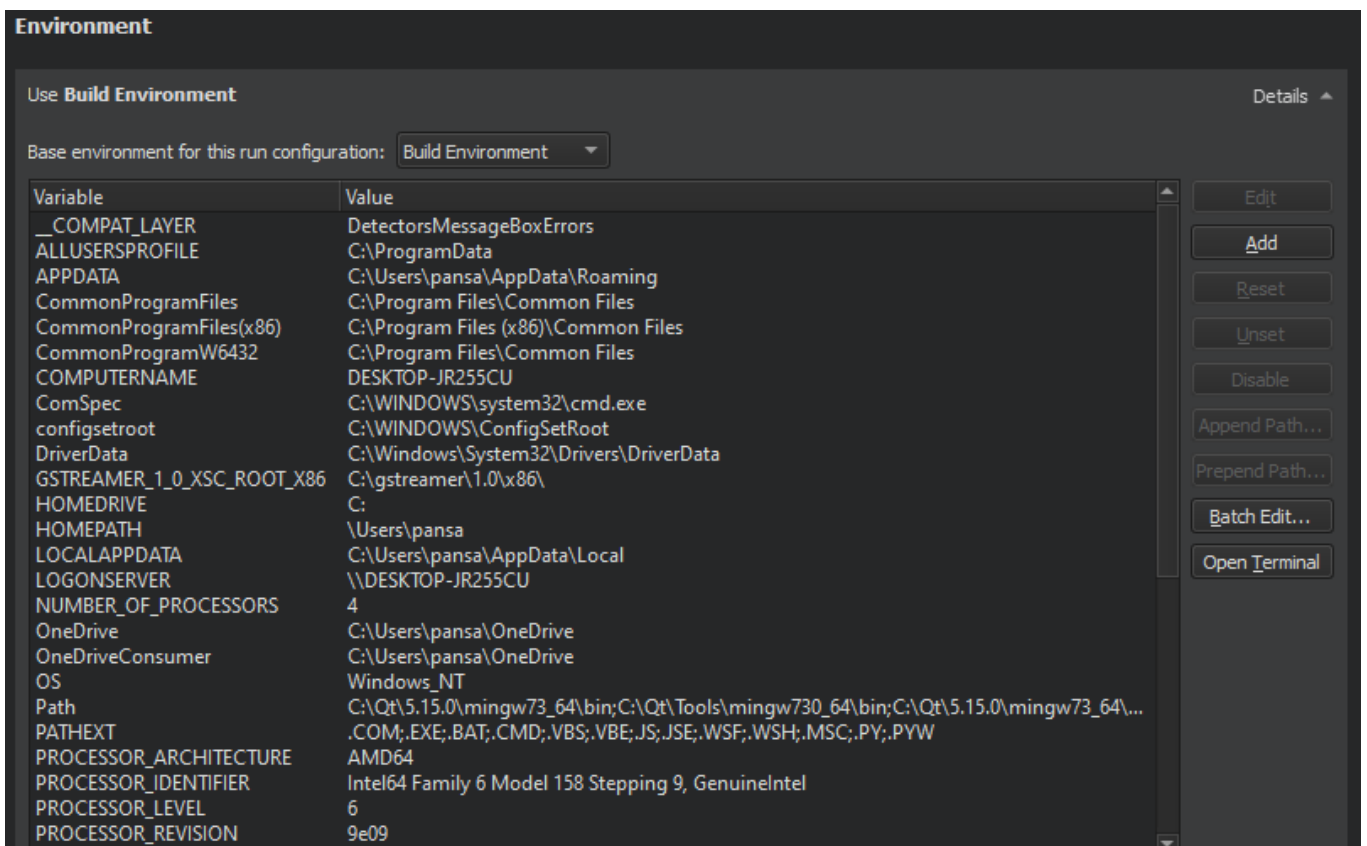
运行设置显示开发主机和设备上可执行文件的路径。

有关部署步骤的详细信息，请参阅[将应用程序部署到 Boot2Qt 设备](#)。

选择运行环境

Qt Creator 根据**设备**类型自动选择用于运行应用程序的环境。您可以在“运行环境”部分中编辑环境或选择其他**环境**。

您可以编辑现有环境变量或添加、重置和取消设置新变量。



细信息，请参阅[构建环境和指定环境设置](#)。

若要在干净的系统环境中运行，请选择“**清理环境**”。

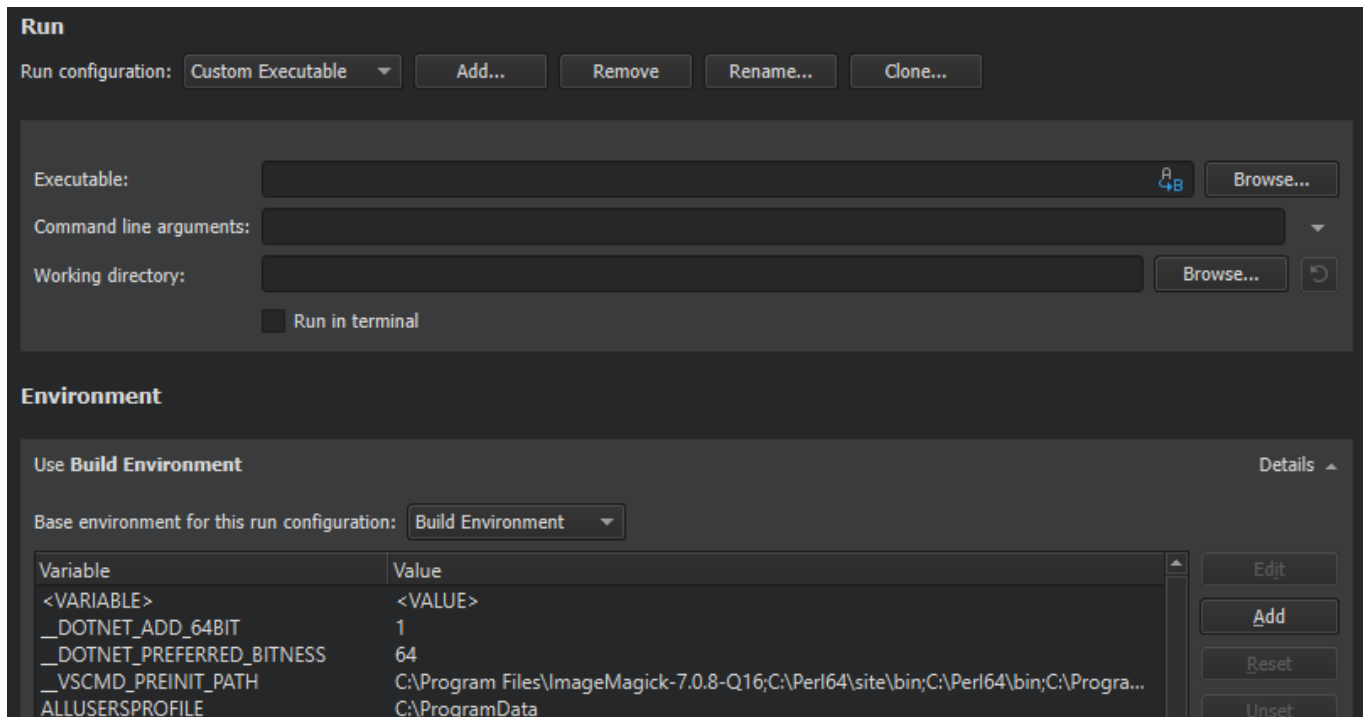
在连接到开发主机的移动设备上运行时，Qt Creator从设备获取有关**设备环境**的信息。通常，编辑设备环境没有意义。

若要修改运行环境的环境变量值，请选择“**批量编辑**”。有关详细信息，请参阅[批量编辑](#)。

Qt创建者手册8.0.2
Topics >

如果您在Qt Creator中使用CMake，Meson或通用项目类型，或者想要运行自定义桌面可执行文件，请为您的项目创建自定义**可执行文件**运行配置。例如，在处理库时，可以运行链接到库的测试应用程序。

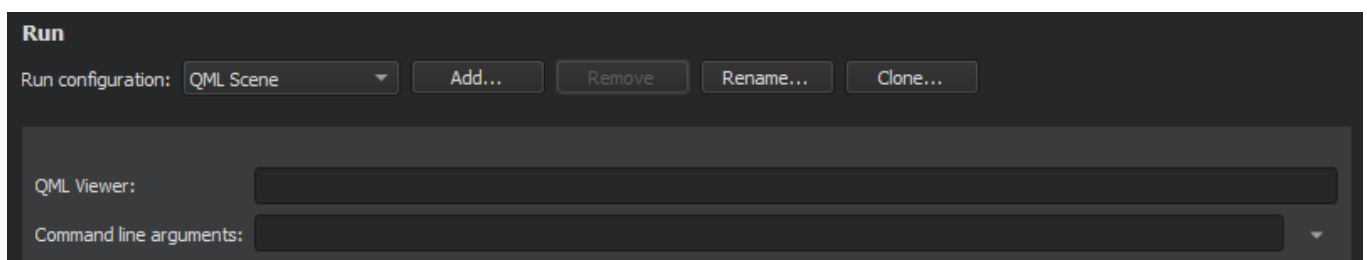
指定要运行的可执行文件、命令行参数、工作目录和要使用的环境变量。



指定Qt快速UI项目的运行设置

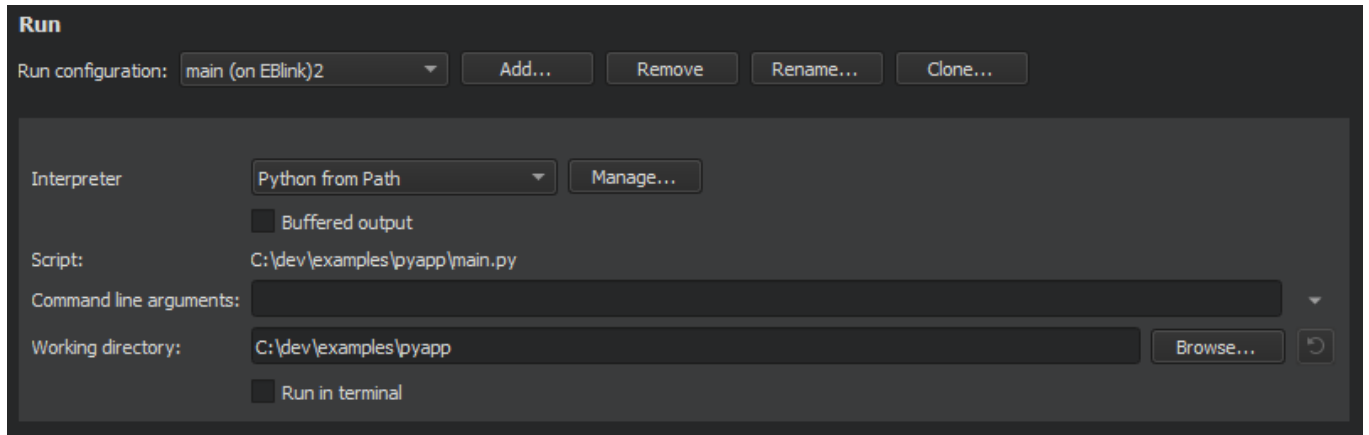
您可以使用**桌面**设备类型为**套件**指定运行设置：

- › 在“QML 查看器”字段中，指定要使用的 QtQML **查看器**。
- › 在 命令行参数 字段中，指定要传递给可执行文件的**参数**。
- › 在**主 QML 文件**中，选择将用于启动 Qt QML 查看器的文件。



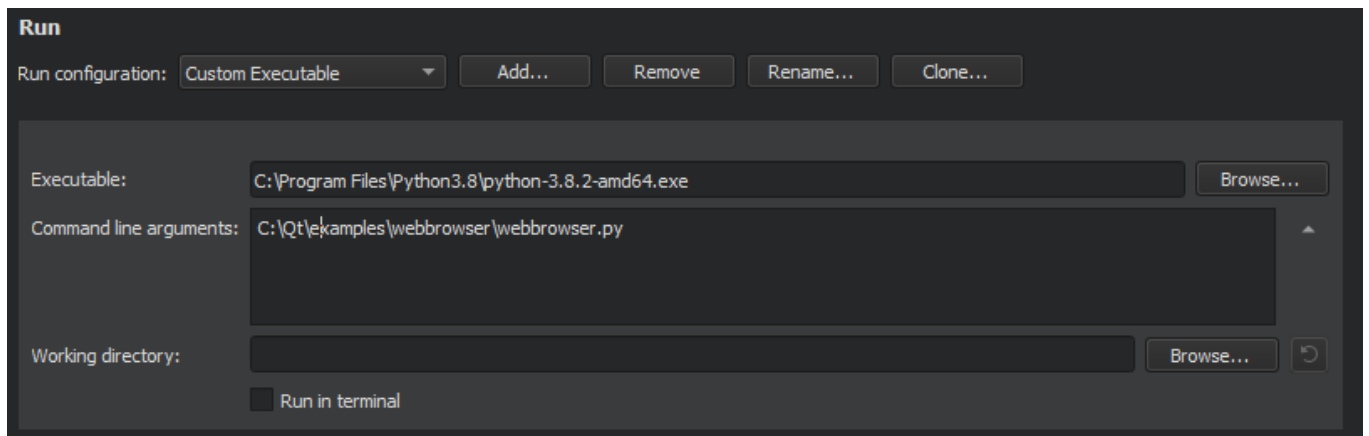
指定 Python 项目的运行设置

您可以指定运行Qt for Python应用程序的设置：



- › 在**解释器**字段中，指定 Python 可执行文件的路径。
- › 选中**缓冲输出**复选框以缓冲输出。这提高了输出性能，但会导致输出延迟。
- › 在**脚本**字段中，可以看到将运行的项目的主文件的路径。
- › 在**命令行参数**字段中，指定要传递给可执行文件的**命令行参数**。

如果要运行其他 Python 文件，请创建自定义可执行运行配置：main.py



1. 选择**添加>自定义可执行文件**。
2. 在**可执行文件**字段中，指定 Python 可执行文件的路径。
3. 在**命令行参数**字段中，选择要运行的 Python 文件。

[指定生成设置](#)

[指定编辑器设置](#)

©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

支持

- 支持服务
- 专业服务
- 合作伙伴
- 训练

社区

- 为Qt做贡献
- 论坛
- 维基
- 下载
- 市场

发牌

- 条款和条件
- 开源
- 常见问题

对于客户

- 支持中心
- 下载
- Qt登录
- 联系我们
- 客户成功案例