

[Qt创作者手册](#) > [调试Qt快速项目](#)

调试Qt快速项目

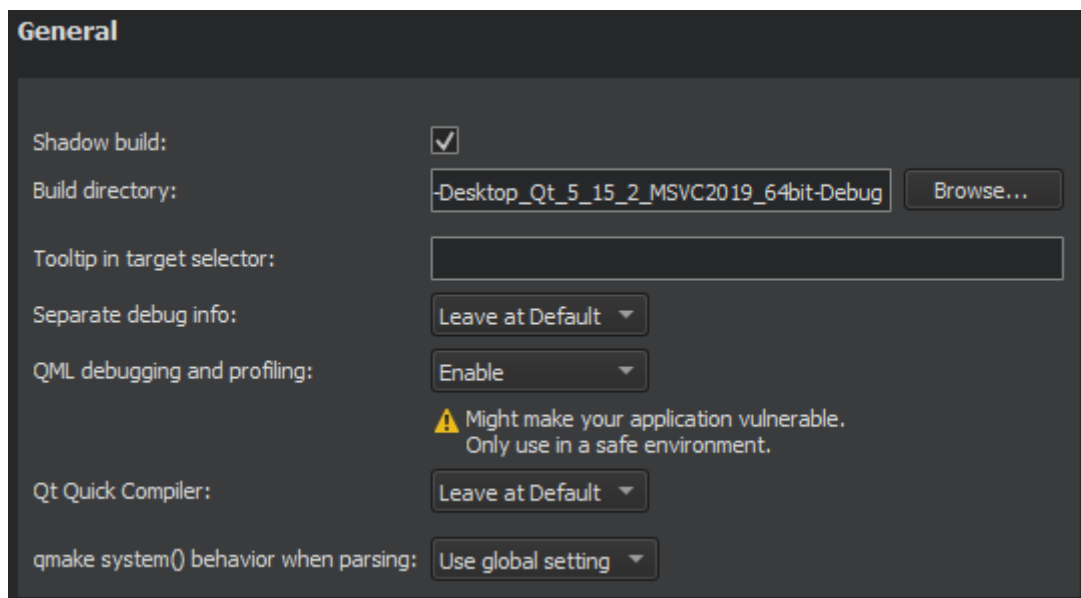
注意：您需要Qt 5.0或更高版本来调试Qt Quick项目。有关如何调试Qt快速项目的示例，请参阅[调试Qt快速示例应用程序](#)。

设置 QML 调试

为Qt Quick项目设置调试的过程取决于[项目的类型](#)：Qt Quick UI或Qt快速应用程序，以及使用的Qt版本。要调试Qt Quick UI项目，请在**项目模式运行**设置的**调试器设置**中选中启用QML复选框。

要调试Qt快速应用程序：

1. 如果使用 qmake 作为构建系统，请确保在“**构建设置**”、“**QML 调试和分析**”字段中启用调试，无论是显式为项目还是默认为全局启用。



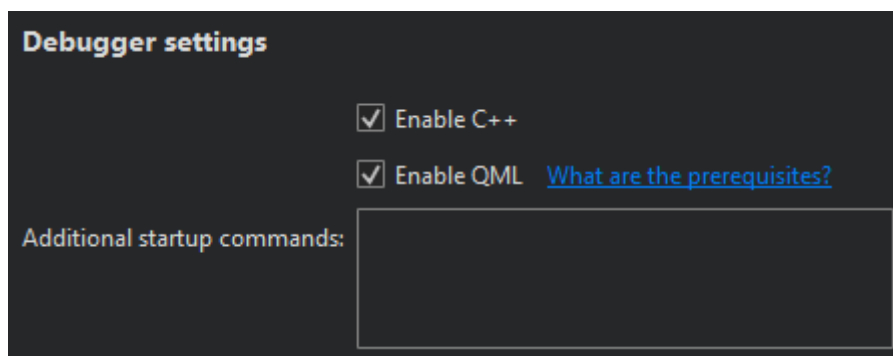
注意：调试需要在 TCP 端口打开套接字，这会带来安全风险。互联网上的任何人都可以连接到您正在调试的应用程序并执行任何JavaScript函数。因此，必须确保端口受到防火墙的正确保护。

2. 在“**运行设置**”->“**调试器设置**”部分中，选中“启用 QML”复选框以**启用 QML**调试。

注意：调试所需的插件在Qt Creator和Qt安装期间自动安装。如果您计划调试 QML 应用程序，请不要删除它们。qmltooling

混合C++/QML 调试

要同时调试应用程序的 C++ 和 QML 部分，请在项目**运行**设置的**调试器设置**部分中选中两种语言的启用C++和启用 QML复选框。



启动 QML 调试

若要启动应用程序，请选择“**调试**>**启动调试**>**启动项目的调试**”或按F5。应用程序开始运行后，其行为和执行将照常进行。然后，您可以执行以下任务：

- › 调试 JavaScript 函数
- › 执行 JavaScript 表达式以获取有关应用程序状态的信息
- › 检查 QML 属性和 JavaScript 变量，并在运行时临时更改它们

调试已在运行的应用程序：

1. 使用适当的配置参数构建应用程序（如果使用Qt Creator构建应用程序，它会自动使用正确的配置）：

- › 使用 CMake 时，`target_compile_definitions`命令在 CMakeLists.txt 文件中定义：
`target_compile_definitions(myapp PRIVATE QT_QML_DEBUG)`

其中`myapp`是要调试的应用程序。

- › 使用 qmake 时，将为 .pro 文件中的`CONFIG`属性定义以下值：`CONFIG += qml_debug`

2. 使用以下参数启动应用程序：

```
-qmljsdebugger=port:<port>[,host:<ip address>][,block]
```

其中（必需）指定调试端口，（可选）指定运行应用程序的主机的 IP 地址，以及（可选）阻止应用程序运行，直到调试客户端连接到服务器。这将启用从头开始调试。`port ip address block`

注意：仅当应用程序以块模式启动时，才能设置断点。

3. 选择**调试**>**启动调试**>**附加到 QML 端口**。

调试 JavaScript 函数

您可以使用Qt Creator**调试模式**在调试时检查应用程序的状态。可以通过以下方式与调试器交互：

- › 设置断点
- › 查看调用堆栈跟踪
- › 查看局部变量和函数参数
- › 计算表达式

检查项目

在应用程序运行时，您可以使用“**局部变量**”视图浏览 QML 项目结构。

Locals		
Name	Value	Type
QQuickView	object	QQuickView
Properties	list	
QQuickRootItem	object	QQuickRootItem
Properties	list	
root	object	Rectangle
Properties	list	
Transition	object	Transition
State	object	State
Properties	list	
changes	<unknown valu...	QQmlListProperty<QQuickStateOperation>
extend		QString
name	in-game	QString
objectName		QString
when		QQmlBinding *
gameOverTimer	object	Timer
Image	object	Image
gameCanvas	object	GameArea
menu	object	Item
scoreBar	object	Image
bottomBar	object	Image
Connections	object	Connections
stateChangeAnim	object	SequentialAnimation
Keys	object	Keys
QQmlEngine	object	QQmlEngine
QQmlFileSelector	object	QQmlFileSelector
Component	object	Component

若要在与调试器交互时使用应用程序保持可见，请选择“**调试>在顶部显示应用程序**”。

您可以通过以下方式在“**局部变量**”视图中查看 QML 项目：

- › 展开对象树中的项。
- › 在代码编辑器中选择该项。
- › 选择“**调试**”>**选择**以激活选择模式，然后单击正在运行的应用程序中的某个项目。

若要临时更改属性值而不编辑源，请双击它们并输入新值。您可以在正在运行的应用程序中查看结果。

在选择模式下，可以单击正在运行的应用程序中的项以跳转到代码中的项定义。所选项的属性将显示在“**局部变量**”视图中。

如果您的应用程序使用的是Qt 5.7或更高版本，或者您的应用程序使用的是Qt的早期版本并且基于该类，则**选择工具**将被启用。您还可以在正在运行的应用程序中查看项目层次结构：QQuickView

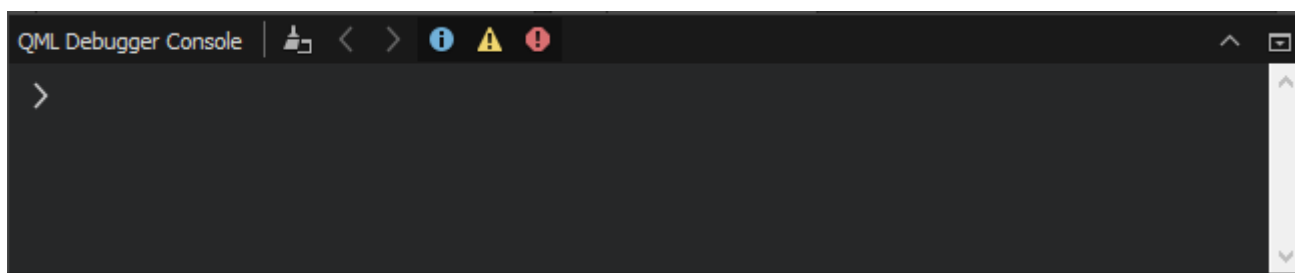
双击正在运行的应用程序中的某个项，以在光标位置循环浏览项堆栈。

要切换出选择模式，请切换“**选择**”菜单项。

要将Qt QML查看器中运行的应用程序移到前面，请选择**调试>在顶部显示应用程序**。

执行 JavaScript 表达式

当应用程序被断点中断时，您可以使用**QML 调试器控制台**在当前上下文中执行 JavaScript 表达式。若要打开它，请选择“**视图>输出>QML 调试器控制台**”。



可以临时更改属性值，而无需编辑源，并在正在运行的应用程序中查看结果。可以在代码中永久更改属性值。

在运行时应用 QML 更改

在**QML 调试器控制台**或“**局部变量**”或“**表达式**”视图中更改属性值时，这些属性值会立即在正在运行的应用程序中更新，但不会在源代码中更新。

[< 使用调试帮助程序](#)

[调试C++示例应用程序 >](#)

©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

品牌



投资者
编辑部
职业
办公地点

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