

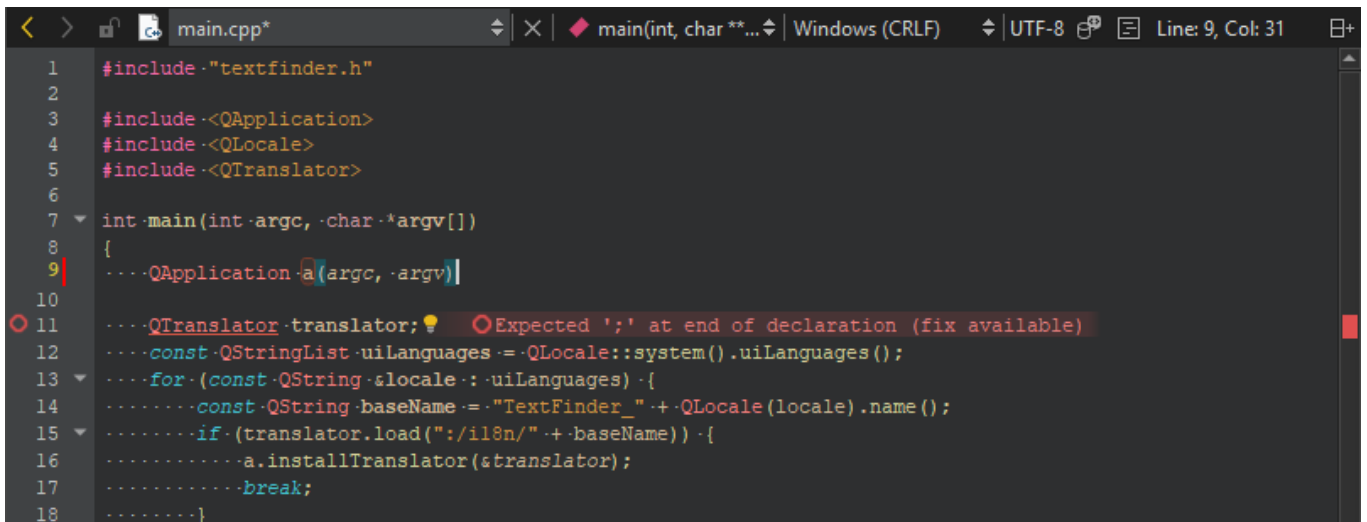
[Qt创作者手册](#) > [检查代码语法](#)

## 检查代码语法

在你编写代码时，Qt Creator会检查代码语法。当Qt Creator在代码中发现语法错误时，它会在错误或图标上为其加下划线，显示一个图标，并显示错误详细信息。如果有可用的修复程序，您可以通过单击来应用它🔧。

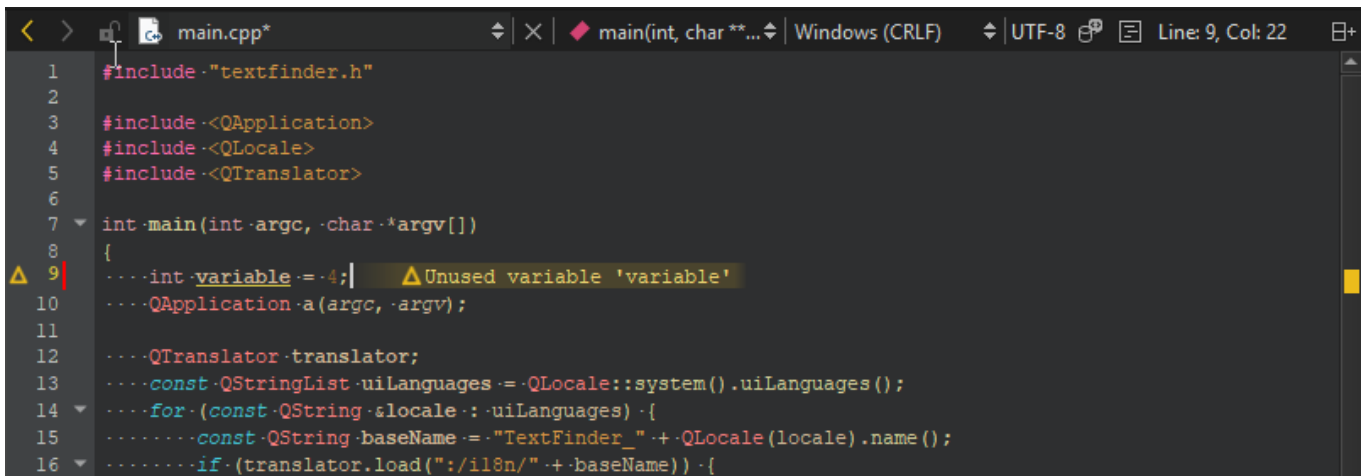
同样，当您处理JavaScript对象表示法（JSON）实体的实例时，Qt Creator会为JSON数据结构中的错误加下划线。

在以下屏幕截图中，Qt Creator 注释了一个错误，因为行尾缺少分号：



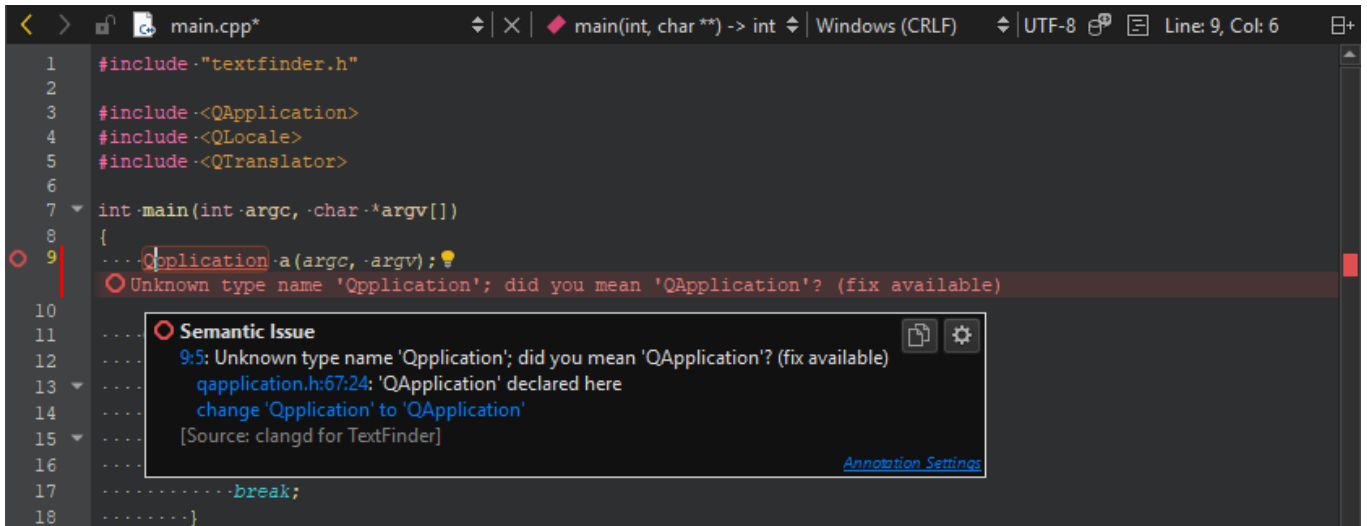
```
< > main.cpp* | main(int, char **... | Windows (CRLF) | UTF-8 | Line: 9, Col: 31
1  #include < "textfinder.h"
2
3  #include < <QApplication>
4  #include < <QLocale>
5  #include < <QTranslator>
6
7  int main(int argc, char *argv[])
8  {
9      QApplication a(argc, argv)
10
11      QTranslator translator; Expected ';' at end of declaration (fix available)
12      QStringList uiLanguages = QLocale::system().uiLanguages();
13      for (const QString &locale : uiLanguages) {
14          const QString &baseName = "TextFinder_" + QLocale(locale).name();
15          if (translator.load(":/i18n/" + baseName)) {
16              a.installTranslator(&translator);
17              break;
18          }
19      }
```

在以下屏幕截图中，Qt Creator 发出警告，因为未使用该变量：



```
< > main.cpp* | main(int, char **... | Windows (CRLF) | UTF-8 | Line: 9, Col: 22
1  #include < "textfinder.h"
2
3  #include < <QApplication>
4  #include < <QLocale>
5  #include < <QTranslator>
6
7  int main(int argc, char *argv[])
8  {
9      int variable = 4; Unused variable 'variable'
10     QApplication a(argc, argv);
11
12     QTranslator translator;
13     QStringList uiLanguages = QLocale::system().uiLanguages();
14     for (const QString &locale : uiLanguages) {
15         const QString &baseName = "TextFinder_" + QLocale(locale).name();
16         if (translator.load(":/i18n/" + baseName)) {
```

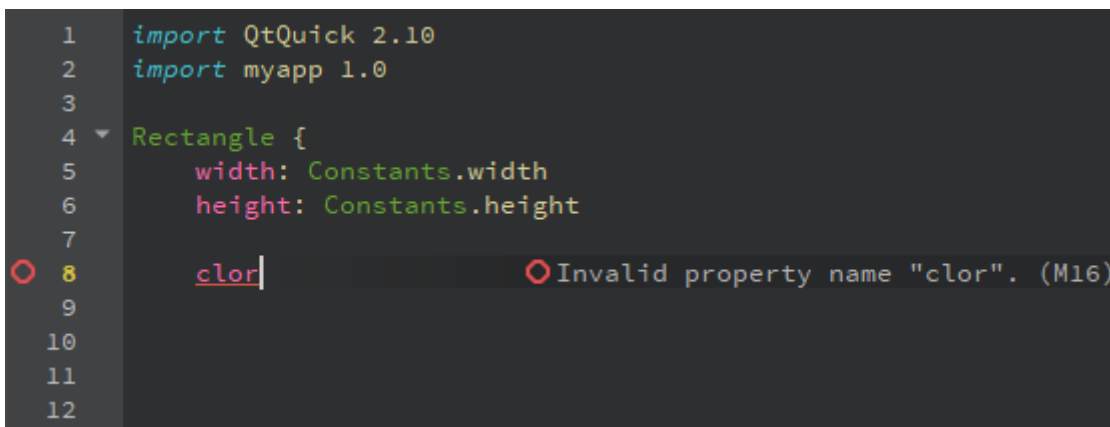
在以下屏幕截图中，Qt 类名包含拼写错误：



您可以使用工具提示弹出窗口中的图标将错误或警告消息复制到剪贴板，或者忽略和隐藏它。选择“批注设置”链接以指定行批注的位置。

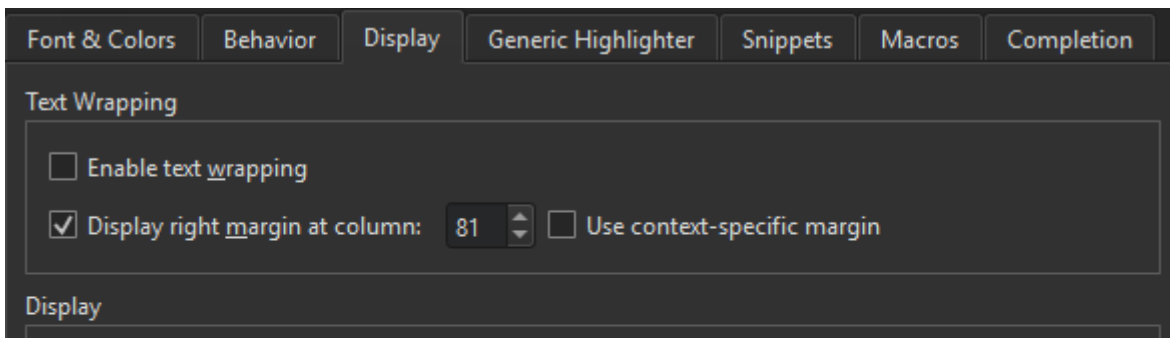
要修改用于为错误和警告添加下划线的颜色，请选择“编辑>首选项>文本编辑器>字体和颜色>复制”，然后为“错误”和“警告”选择新颜色。

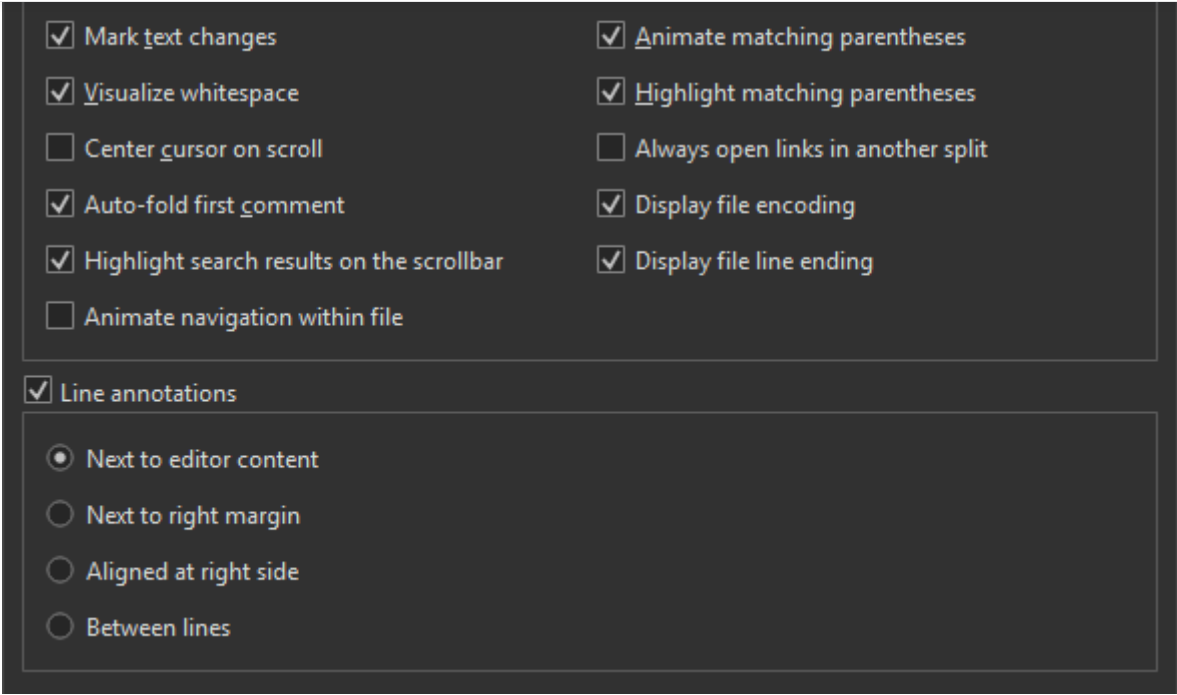
在编写 QML 或 JavaScript 代码时，注释可能包含 JavaScript 和 QML 错误代码中列出的错误代码。



## 指定线注释位置

若要指定批注的显示位置，请选择“编辑>首选项>文本编辑器>显示>行批注”，然后选择是直接在代码旁边、与代码右侧对齐还是在右边距中显示批注。如果通常没有足够的空间在文本旁边显示批注，则在行之间显示批注会很有用。





如果通过取消选中该复选框来隐藏批注，则可以将鼠标指针移到图标上以查看它们。

## JavaScript 和 QML 错误代码

您可以对项目中的 QML 和 JavaScript 代码运行静态检查，以查找常见问题。若要运行检查，请选择“**工具** > QML/JS > **运行检查**”或按Ctrl+Shift+C。结果显示在“**问题**”中的“**QML**”和“**QML分析**”筛选器中。

许多错误消息类似于 Douglas Crockford 的 [JSLint](#) 工具中的错误消息。有关 JSLint 错误的更多信息，请参见 [JSLint 错误说明](#)。

编号	严厉	消息	描述
M1	错误	枚举的值无效	
M2	错误	枚举值必须是字符串或数字	
M3	错误	预期数值	
M4	错误	预期的布尔值	
M5	错误	预期的字符串值	
M6	错误	网址无效	
M7	警告	文件或目录不存在	
M8	错误	颜色无效	
M9	错误	无效的转义序列	

	误		
M11	错误	预期标识	另请参阅: <a href="#">期望标识符, 而是看到“{a}” (保留字)</a> 。
M14	错误	无效的 ID	另请参阅: <a href="#">期望标识符, 而是看到“{a}” (保留字)</a> 。
M15	错误	重复的 ID	文件中的 ID 必须是唯一的。另请参阅: <a href="#">重复键“{a}”</a> 。
M16	错误	属性名称无效name	
M17	错误	Name没有成员	
M18	错误	Field不是object	
M19	警告	条件中的分配	这可能是键入错误。如果是有意的, 请将作业括在括号中。
M20	警告	未终止的非空案例块	案例块应为空或以流控制语句 (如, 或) 结尾。或者, 您可以通过以注释结尾来指示故意失败。breakreturncontinue// fall through
M23	警告	请勿使用eval	另请参阅: <a href="#">eval is evil</a> 。
M28	警告	遥不可及	指示永远不会执行带下划线的语句。
M29	警告	请勿使用with	另请参阅: <a href="#">意外的“与”</a> 。
M30	警告	不要使用逗号表达式	
M31	警告	不必要的消息抑制	
M103	警告	Name已经是一个形式参数	
M104	警告	Name已经是一个函数	
M105	警告	声明前使用的变量name	
M106	警告	Name已经是一个变量	
M107	警告	Name被声明多次	在函数中声明的变量在函数中的任何地方始终可见, 即使在嵌套块或语句条件中声明也是如此。重新声明变量不起作用。for
M108	警告	函数在其声明之前使用name	另请参阅: <a href="#">{a} 在定义之前就已使用</a> 。
M109	警告	不要用作构造函数 Boolean	另请参阅: <a href="#">不要使用 {a} 作为构造函数</a> 。

M111	警告	不要用作构造函数Object	另请参阅: <a href="#">不要使用 {a} 作为构造函数。</a>
M112	警告	不要用作构造函数Array	另请参阅: <a href="#">不要使用 {a} 作为构造函数。</a>
M113	警告	不要用作构造函数Function	另请参阅: <a href="#">不要使用 {a} 作为构造函数。</a>
M114	提示	关键字和左括号应用单个空格分隔function	另请参阅: <a href="#">预计 {a} 和 {b} 之间正好有一个空格。</a>
M115	警告	不要使用独立块	块不会影响变量范围。因此, 与 <code>ifwhile</code> 等无关的块没有效果, 应避免使用。
M116	警告	不要使用空表达式	
M117	警告	令人困惑的优点	另请参阅: <a href="#">令人困惑的加号。</a>
M119	警告	令人困惑的缺点	另请参阅: <a href="#">令人困惑的缺点。</a>
M121	提示	在一行上声明所有函数变量	
M123	提示	不必要的括号	
M126	警告	<code>==</code> 并可能执行类型强制, 使用或避免它 <code>!===!==</code>	允许非严格相等比较将其参数转换为通用类型。这可能会导致意想不到的结果, 例如真实。使用严格相等运算符并明确说明所需的转换。 <code>' \t\r\n' == 0===!==</code>
M127	警告	表达式语句应仅为赋值、调用或删除表达式	
M128	错误	状态不能具有指定的子项	
M129	错误	类型不能递归实例化	
M201	提示	将 <code>var</code> 声明放在函数的开头	另请参阅: <a href="#">将“var”声明移动到函数顶部。</a>
M202	提示	每行仅使用一个语句	
M220	错误	不支持此类型 (类型名称) 作为 UI 文件 (.ui.qml) 的根元素	有关支持的 QML 类型的更多信息, 请参阅 <a href="#">UI 文件</a> 。
M221	错误	UI 文件 (.ui.qml) 不支持此类型 (类型名称)	有关支持的 QML 类型的更多信息, 请参阅 <a href="#">UI 文件</a> 。
M222	错误	UI 文件 (.ui.qml) 中不支持函数	有关支持的 JavaScript 函数的列表, 请参阅 <a href="#">支持的方法</a> 。
M223	错误	UI 文件 (.ui.qml) 不支持 JavaScript 块	有关支持的功能的详细信息, 请参阅 <a href="#">UI 文件</a> 。
M224	错误	UI 文件 (.ui.qml) 不支持	有关支持的 QML 类型的更多信息, 请参阅 <a href="#">UI 文件</a> 。

	错误	根项下无子项	
M226	错误	UI 文件 (.ui.qml) 不支持引用根项的父项	有关支持的功能的详细信息，请参阅 <a href="#">UI 文件</a> 。
M227	错误	不要在 UI 文件 (.ui.qml) 中混合使用翻译函数	即使支持几种不同的翻译函数（例如 <code>andare</code> ），您也应该选择一个并在 UI 文件中一致地使用它。有关更多信息，请参阅 <a href="#">Qt QML 方法</a> 。 <code>qsTrqsTrId</code>

Qt创建者手册8.0.2

Topics >

M301	错误	无法解析原型 <code>nameobject</code>	
M302	错误	无法解析原型 <code>name</code>	
M303	错误	原型循环，最后一个非重复组件是 <code>name</code>	
M304	错误	属性类型无效 <code>name</code>	
M305	警告	<code>==</code> 并执行类型强制，使用或避免它！ <code>==== !==</code>	参见 <a href="#">M126</a> 。
M306	警告	调用以大写字母开头的函数应使用 <code>new</code>	按照惯例，以大写字母开头的函数是只应与之一起使用的构造函数。 <code>new</code>
M307	警告	仅用于以大写字母开头的函数 <code>new</code>	另请参阅： <a href="#">不要使用 {a} 作为构造函数</a> 。
M308	警告	不要用作构造函数 <code>Number</code>	另请参阅： <a href="#">不要使用 {a} 作为构造函数</a> 。
M309	提示	在二元运算符周围使用空格	
M310	警告	无意的空块，使用 <code>{}</code> 表示空对象文字	
M311	提示	使用代替 <code>oforto</code> 提高性能 <code>typevarvariant</code>	
M312	错误	缺少属性 <code>number</code>	
M313	错误	预期对象值	
M314	错误	预期的数组值	
M315	错误	<code>Value</code> 预期值	
M316	错误	最大数值为 <code>number</code>	
M317	错误	最小数值为 <code>number</code>	
M318	错误	最大数值是独占的	

	误		
M320	错误	字符串值与所需模式不匹配	
M321	错误	最小字符串值长度为 number	
M322	错误	最大字符串值长度为 number	
M323	错误	Number 数组值中预期的元素	另请参阅： <a href="#">数组文字表示法</a> <a href="#">[]</a> 更可取。
M324	错误	达到访问 AST 的最大递归限制，代码模型将不可靠且很可能无效	
M325	警告	逻辑值不依赖于实际值	
M326	错误	组件只允许有一个子元素	更多信息，请参见 <a href="#">组件</a> 。
M327	警告	组件需要子元素	更多信息，请参见 <a href="#">组件</a> 。
M400	警告	重复导入	导入语句已添加两次。有关更多信息，请参阅 <a href="#">导入语句</a> 。

## 重置代码模型

如果在代码编辑器中打开 QML 文件时更改构建和运行工具包，则代码模型可能会损坏。以下错误消息表明可能已经发生这种情况：*使用 Qt Quick 1 代码模型而不是 Qt Quick 2。*

当您将鼠标指针移到代码编辑器中 Qt Creator 下划线的代码上或在 Qt Design Studio 中打开 QML 文件时，可以看到错误消息。

若要重置代码模型，请选择“**工具>QML/JS>重置代码模型**”。

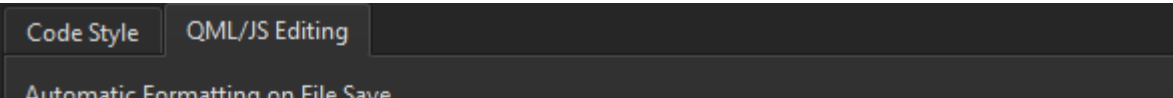
如果这没有帮助，请尝试将 QML 仿真层更改为使用与构建和运行工具包中选择的相同 Qt 版本构建的仿真层。

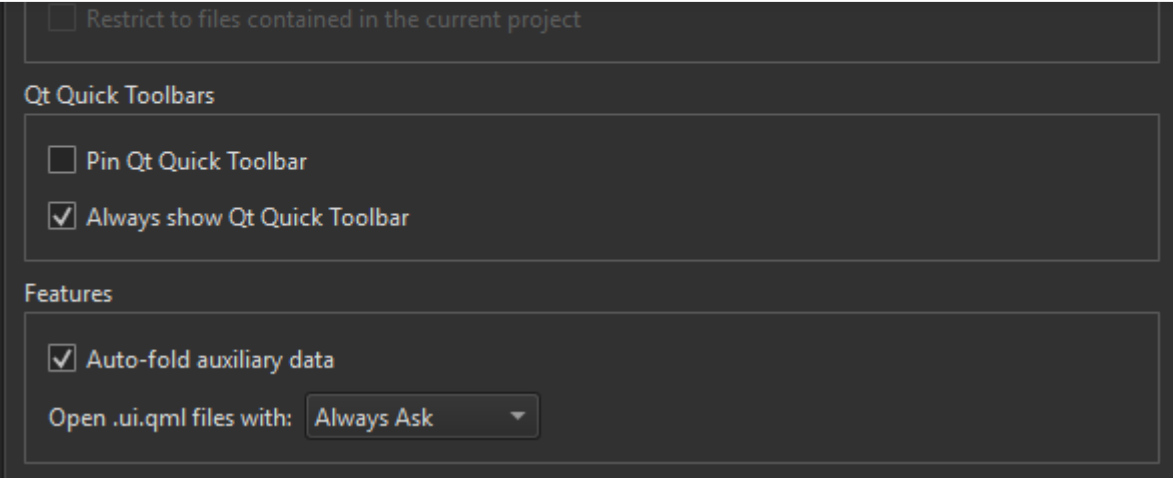
## 检查 QML 和 JavaScript

要检查 QML 和 JavaScript 属性、方法和枚举，请将光标移到它们上，然后在光标下选择**Tools>QML/JS>Inspect API for Element**。

## 自动格式化 QML/JS 文件

要在保存时自动格式化 QML/JS 文件，请选择编辑>**首选项>Qt 快速>QML/JS 编辑>启用文件保存时自动格式化**。





[< 语义突出显示](#)

[完成代码 >](#)

©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



论坛  
维基  
下载  
市场

© 2022 Qt公司

[反馈](#) [登录](#)