



豆子空间

<http://devbean.blog.51cto.com> [【复制】](#) [【订阅】](#)

[博 客](#) | [图 库](#) | [写 博 文](#) | [帮 助](#)

[首 页](#) | [Java](#) | [JavaScript](#) | [C/C++](#) | [Qt](#) | [Flex](#) | [Photoshop](#) | [Firefox](#) | [杂 谈](#)

FinderCheng 的BLOG

相关视频课程

更多



[写留言](#)

[去学院学习](#)

[发消息](#)

[加友情链接](#)

[进家园](#) [加好友](#)



精通Spark内核系列课程5：核心源码剖析(共309人学习)



【企业面试必考技术】跟着老男孩学习运维必2385人学习



Windows 高级编程（核心编程 API编程）（上1912人学习

[博主的更多文章>>](#)

原创 Qt核心剖析：moc

2010-07-22 13:54:01

标签: [核心](#) [源码](#) [Qt](#) [休闲](#) [剖析](#)

原创作品，允许转载，转载时请务必以超链接形式标明文章 [原始出处](#)、作者信息和本声明。否则将追究法律责任。
任。<http://devbean.blog.51cto.com/448512/355100>

博客统计信息

[51CTO推荐博客](#)

用户名: FinderCheng

文章数: 123

评论数: 989

访问量: 3542474

无忧币: 2042

博客积分: 3388

博客等级: 7

注册日期: 2008-08-11

热门专题

[更多>>](#)

前面我们说过，Qt 不是使用的“标准的” C++ 语言，而是对其进行了一定程度的“扩展”。这里我们从Qt新增加的关键字就可以看出来：signals、slots 或者 emit。所以有人会觉得 Qt 的程序编译速度慢，这主要是因为 在 Qt 将源代码交给标准 C++ 编译器，如 gcc 之前，需要事先将这些扩展的语法去除掉。完成这一操作的就是 moc。

moc 全称是 Meta-Object Compiler，也就是“元对象编译器”。Qt 程序在交由标准编译器编译之前，先要使用 moc 分析 C++ 源文件。如果它发现在一个头文件中包含了宏 Q_OBJECT，则会生成另外一个 C++ 源文件。这个源文件中包含了 Q_OBJECT 宏的实现代码。这个新的文件名字将会是原文件名前面加上 moc_ 构成。这个新的文件同样将进入编译系统，最终被链接到二进制代码中去。因此我们可以知道，这个新的文件不是“替换”掉旧的文件，而是与原文件一起参与编译。另外，我们还可以看出一点，moc 的执行是在预处理器之前。因为预处理器执行之后，Q_OBJECT 宏就不存在了。

既然每个源文件都需要 moc 去处理，那么我们在什么时候调用了它呢？实际上，如果你使用 qmake 的话，这一步调用会在生成的 makefile 中展现出来。从本质上来说，qmake 不过是一个 makefile 生成器，因此，最终执行还是通过 make 完成的。

为了查看 moc 生成的文件，我们使用一个很简单的 cpp 来测试：

热门文章

- Qt学习之路(17): Qt标准..
- Qt学习之路(2): Hello, w..
- Qt学习之路(1): 前言
- Qt学习之路(50): QString
- Qt学习之路(7): 创建一个..
- Qt学习之路(8): 创建一个..
- Qt学习之路(15): Qt标准..
- 深入Java单例模式

搜索BLOG文章

搜索

最近访客



10552..



72974..



15192..



jim003



qinzy212



61940..



sucha..



肇庆..



一心永成



a9341..



loserfly



gaoku..

最新评论

飘雪夏娜样: 书上说的都说了, 书上没有讲清楚的..

test.cpp

```
01. class Test : public QObject
02. {
03.     Q_OBJECT
04. public:
05.     explicit Test(QObject *parent = 0);
06. signals:
07. public slots:
08.     };
```

这是一个空白的类, 什么都没有实现。在经过编译之后, 我们会在输出文件夹中找到 moc_test.cpp:

moc_test.cpp

```
01. /*****
02. ** Meta object code from reading C++ file 'test.h'
03. **
04. ** Created: Thu Jul 22 13:06:45 2010
05. **      by: The Qt Meta Object Compiler version 62 (Qt 4.6.3)
06. **
07. ** WARNING! All changes made in this file will be lost!
08. *****/
09.
10. #include "../test.h"
11. #if !defined(Q_MOC_OUTPUT_REVISION)
12. #error "The header file 'test.h' doesn't include <QObject>."
13. #elif Q_MOC_OUTPUT_REVISION != 62
14. #error "This file was generated using the moc from 4.6.3. It"
15. #error "cannot be used with the include files from this version of Qt."
16. #error "(The moc has changed too much.)"
17. #endif
18.
19. QT_BEGIN_MOC_NAMESPACE
20. static const uint qt_meta_data_Test[] = {
21.
22.     // content:
23.     4,          // revision
24.     0,          // classname
25.     0,    0,   // classinfo
26.     0,    0,   // methods
27.     0,    0,   // properties
28.     0,    0,   // enums/sets
29.     0,    0,   // constructors
30.     0,         // flags
31.     0,         // signalCount
32.
33.     0          // eod
34. };
35.
36. static const char qt_meta_stringdata_Test[] = {
37.     "Test\0"
38. };
39.
40. const QMetaObject Test::staticMetaObject = {
41.     { &QObject::staticMetaObject, qt_meta_stringdata_Test,
42.       qt_meta_data_Test, 0 }
43. };
44.
45. #ifndef Q_NO_DATA_RELOCATION
46. const QMetaObject &Test::getStaticMetaObject() { return staticMetaObject; }
47. #endif //Q_NO_DATA_RELOCATION
48.
```

aguai1617: 请问有没有从应用程序拖动文件到本..

wx275271279: “这里其实是一个冗余的操作，因为..

wx275271279: 另外还有一点，在一些必须精确操作..

wx275271279: 讲的深入透彻，楼主厉害，十分感谢

51CTO推荐博文 更多>>

- 运维角度浅谈MySQL数据库优化
- Oracle Study之一--AMD CPU安装Orac..
- svn利用钩子脚本功能实现代码同步..
- 【UNITY3D 游戏开发之八】Unity编..
- PL/SQL中如何让程序每隔几秒插入..
- Keepalived+LVS+MariaDB Galera C..
- Ubuntu14.04快速搭建SVN服务器及..
- 整理的部分Java和C#不同点
- Apache Httpd服务器之虚拟机详解
- 将Uhost上的MySQL迁移到UDB
- mysql分布式中间件cobar

友情链接

- 石榴石网
- 趣闻网
- 汉堡店加盟
- 奶茶店加盟
- IT精品课程
- 我的主页
- PicWorks
- Qt 文档翻译
- Exchange 2010
- 51CTO博客开发

```
49.  const QMetaObject *Test::metaObject() const
50.  {
51.      return QObject::d_ptr->metaObject ? QObject::d_ptr->metaObject : &staticMetaObject;
52.  }
53.
54.  void *Test::qt_metacast(const char *_cname)
55.  {
56.      if (!_cname) return 0;
57.      if (!strcmp(_cname, qt_meta_stringdata_Test))
58.          return static_cast<void*>(const_cast< Test*>(this));
59.      return QObject::qt_metacast(_cname);
60.  }
61.
62.  int Test::qt_metacall(QMetaObject::Call _c, int _id, void **_a)
63.  {
64.      _id = QObject::qt_metacall(_c, _id, _a);
65.      if (_id < 0)
66.          return _id;
67.      return _id;
68.  }
69.  QT_END_MOC_NAMESPACE
```

可以看到，moc_test.cpp 里面为 Test 类增加了很多函数。然而，我们并没有实际写出这些函数，它是怎么加入类的呢？别忘了，我们还有 Q_OBJECT 这个宏呢！在 qobjectdefs.h 里面，找到 Q_OBJECT 宏的定义：

```
01.  #define Q_OBJECT \
02.  public: \
03.      Q_OBJECT_CHECK \
04.      static const QMetaObject staticMetaObject; \
05.      Q_OBJECT_GETSTATICMETAOBJECT \
06.      virtual const QMetaObject *metaObject() const; \
07.      virtual void *qt_metacast(const char *); \
08.      QT_TR_FUNCTIONS \
09.      virtual int qt_metacall(QMetaObject::Call, int, void **); \
10.  private:
```

这下了解了：正是对 Q_OBJECT 宏的展开，使我们的 Test 类拥有了这些多出来的属性和函数。注意，QT_TR_FUNCTIONS 这个宏也是在这里定义的。也就是说，如果你要使用 tr() 国际化，就必须使用 Q_OBJECT 宏，否则是没有 tr() 函数的。这期间最重要的就是 virtual const QMetaObject *metaObject() const; 函数。这个函数返回 QMetaObject 元对象类的实例，通过它，你就获得了 Qt 类的反射的能力：获取本对象的类型之类，而这一切，都不需要 C++ 编译器的 RTTI 支持。Qt 也提供了一个类似 C++ 的 dynamic_cast() 的函数 qobject_case()，而这一函数的实现也不需要 RTTI。另外，一个没有定义 Q_OBJECT 宏的类与它最接近的父类是同一类型的。也就是说，如果 A 继承了 QObject 并且定义了 Q_OBJECT，B 继承了 A 但没有定义 Q_OBJECT，C 继承了 B，则 C 的 QMetaObject::className() 函数将返回 A，而不是本身的名字。因此，为了避免这一问题，所有继承了 QObject 的类都应该定义 Q_OBJECT 宏，不管你是不是使用信号槽。

本文出自 “豆子空间” 博客，请务必保留此出处http://devbean.blog.51cto.com/448512/355100

分享至:

收藏 +



微信



关注51CTO博客微信
有机会赢下载VIP会员

微信号：blog51cto

上一篇 Qt核心剖析：信息隐藏(3) 下一篇 参与 Qt 文档翻译项目



相关文章

- Qt核心剖析：信息隐藏(3)
- Qt核心剖析：寻找 QObject 的源代码
- Qt4. 4. 3自带教程-这次分成多个文件
- Qt学习之路(7)：创建一个对话框(上)
- 在某网站下的源码里清除的后门代码
- Qt核心剖析：信息隐藏(2)
- Qt核心剖析：信息隐藏(1)
- 建立软件质量保证方法论的核心手段 — 要素..
- Qt学习之路(39)：QListWidget

文章评论

[1楼]  [匿名]beyondzoro 回复

2010-08-18 17:27:18

感谢楼主的分享，热切期待后续的文章！

[2楼]  hxcan 回复

2010-11-26 16:09:48

在不使用Q_OBJECT宏的情况下，可以包含<QObject>，再使用QObject::tr() 函数来实现翻译。

[3楼]楼主  FinderCheng 回复

2010-11-26 16:43:07

回复 hxcan: [2楼]
是的，使用 QObject::tr() 也可以实现翻译的。有时候会强调使用 signal 和 slot 的时候必须使用 Q_OBJECT 宏，忽略了翻译的时候也是需要的，所以很可能这里会有问题。

[4楼]  刘飞1 回复

2013-04-14 23:18:36

问个问题，我安装完成sdk以及vs插件之后，用VS编译designer工程（在tools\designer\目录下）的时候，出现如下错误：
error C2027: 使用了未定义类型 “QtResourceViewPrivate”

经过检查，发现这个类定义在qtresourceview.cpp里面。
检查了头文件，也没有前向引用，moc生成的文件moc_qtresourceview_p.cpp里面调用了这个类，这种情况肯定编译不过的。
大牛能否帮忙看看这个问题是什么情况？ 谢谢

[5楼]  bdss58 回复

2015-01-09 17:19:18

今天被面试官问到这个问题，被问惨了。主要是之前对moc没有概念啊。

发表评论 [51CTO学院2周年庆，分享故事赢大礼包](#)

昵 称:

验证码: 请点击后输入验证码 [博客过2级，无需填写验证码](#)

内 容: