



[\[公告\]](#)

[一去、二三里](#) | [个人中心](#) | [发博文](#) | [消息](#)



一去、二三里的博客

http://blog.sina.com.cn/liang19890820

[首页](#) | [博文目录](#) | [图片](#) | [关于我](#)



[页面设置](#)

[个人中心](#)

[个人资料](#) [\[管理\]](#)



一去、二三里





博客等级: **18**

博客积分: **78**

博客访问: **407,846**

关注人气: **378**

获赠金笔: **12**

赠出金笔: **0**

荣誉徽章: 

[相关博文](#)

[更多>>](#)

[推荐博文](#)

▪ 2550mAh电

▪ 日本雾霾之战: 民众与政府的博弈

▪ 南非猎豹冒死捕杀豪猪被扎满嘴刺

▪ 别光顾着把枪口对准柴静

▪ 马来西亚男子与眼镜王蛇接吻(图

▪ 【早评】降息利好将会在后市中逐

▪ 哪些人能柴静的“穹顶之下”赚

▪ “深圳机场撞人事件”不是一个人

▪ 政采剔除国外品牌: 早该说不了!

▪ 【DIY】做一枚宫灯迎元宵



苗族情人节捕鱼比赛



马语者表演驯马神技

正文

字体大小: [大](#) [中](#) [小](#)

Qt之日志输出文件  (2013-12-04 17:43:18) [\[编辑\]](#)[\[删除\]](#)

[+ 转载](#) [▼](#)

标签: qt qinstallmessagehandl qinstallmsghandler qt日志 qdebug 分类: Qt

在Qt开发过程当中经常使用qDebug等一些输出来调试程序，但是到了正式发布的时候，都会被注释或者删除，采用日志输出来代替。

做过项目的童鞋可能都使用过日志功能，以便有异常错误能够快速跟踪、定位，Qt也提供的类似的机制。之前用Qt4做项目时使用的是Qt::qInstallMsgHandler(), 到了Qt5，使用了新的Qt::qInstallMessageHandler()来替代，详情请查看Qt助手（C++ API changes）。

描述

助手中在C++ API changes中提到: Qt::QDebug(), Qt::qWarning(), Qt::qCritical(), Qt::qFatal()被改变为宏来跟踪源代码的消息来源。被打印的信息可以被配置（用于缺省消息处理程序），通过设置该新的环境变量QT_MESSAGE_PATTERN。Qt::qInstallMsgHandler()已过时，因此建议使用Qt::qInstallMessageHandler()来代替。

级别

- qDebug: 调试信息
- qWarning: 警告信息
- qCritical: 严重错误
- qFatal: 致命错误

官方示例

Example:

```
#include <qapplication.h>
#include <stdio.h>
#include <stdlib.h>

void myMessageOutput(QtMsgType type, const QMessageLogContext &context, const QString &msg)
{
    QByteArray localMsg = msg.toLocal8Bit();
    switch (type) {
        case QtDebugMsg:
            fprintf(stderr, "Debug: %s (%s:%u, %s)\n", localMsg.constData(), context.file, context.line, context.function);
            break;
        case QtWarningMsg:
            fprintf(stderr, "Warning: %s (%s:%u, %s)\n", localMsg.constData(), context.file, context.line, context.function);
            break;
        case QtCriticalMsg:
            fprintf(stderr, "Critical: %s (%s:%u, %s)\n", localMsg.constData(), context.file, context.line, context.function);
            break;
        case QtFatalMsg:
            fprintf(stderr, "Fatal: %s (%s:%u, %s)\n", localMsg.constData(), context.file, context.line, context.function);
            abort();
    }
}

int main(int argc, char **argv)
{
    qInstallMessageHandler(myMessageOutput);
    QApplication app(argc, argv);
    ...
    return app.exec();
}
```

需求（目的）：生成log.txt日志文件，记录详细日志信息（包括等级、所在文件、所在行号、描述信息、产生时间等），以便于快速跟踪、定位。

```
void outputMessage(QtMsgType type, const QMessageLogContext &context, const QString &msg)
{
    static QMutex mutex;
    mutex.lock();
```

1/3



母狮惨遭水牛顶伤



抓拍鲨鱼呲牙怪笑照



曼妙旅程:潜拍海底世界



嬉戏野兔击掌求偶

查看更多>>

谁看过这篇博文

	sftctrl	16分钟前
	ants	3月2日
	furui1234...	3月1日
	柒小希要...	2月28日
	todaysky	2月25日
	马尚津	2月25日
	吾含逸	2月19日
	桔子花开	2月10日
	jackbackm...	2月9日
	BirdZhang	2月7日
	HybridBla...	2月5日
	枇杷大人	2月4日

```
QString text;
switch(type)
{
case QtDebugMsg:
    text = QString("Debug:");
    break;

case QtWarningMsg:
    text = QString("Warning:");
    break;

case QtCriticalMsg:
    text = QString("Critical:");
    break;

case QtFatalMsg:
    text = QString("Fatal:");
}

QString context_info = QString("File:(%1) Line:
(%2)").arg(QString(context.file)).arg(context.line);
QString current_date_time = QDateTime::currentDateTime().toString("yyyy-MM-dd hh:mm:ss ddd");
QString current_date = QString("(%1)").arg(current_date_time);
QString message = QString("%1 %2 %3
%4").arg(text).arg(context_info).arg(msg).arg(current_date);

QFile file("log.txt");
file.open(QIODevice::WriteOnly | QIODevice::Append);
QTextStream text_stream(&file);
text_stream << message << "\r\n";
file.flush();
file.close();

mutex.unlock();
}

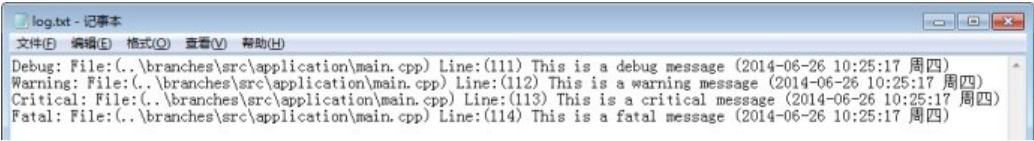
int main(int argc, char *argv[])
{
    QApplication app(argc, argv);

    //注册MessageHandler
    qInstallMessageHandler(outputMessage);

    //打印日志到文件中
    qDebug("This is a debug message");
    qWarning("This is a warning message");
    qCritical("This is a critical message");
    qFatal("This is a fatal message");

    return app.exec();
}
```

效果如下:



注：
技术在于交流、沟通，转载请注明出处并保持作品的完整性。
作者：`☆奋斗ing♥孩子` 原文：http://blog.sina.com.cn/s/blog_a6fb6cc90101guc5.html。



分享：

阅读 (3296) | 评论 (3) | 收藏 (0) | 已有1人转载▼ | 喜欢▼ | 打印 已投稿到： 排行榜

前一篇：QCompleter自动补全
后一篇：Qt之日志输出窗口

评论 重要提示：警惕虚假中奖信息 [发评论]

ajun



感谢，问题就在于这个qt升级到5, 解决了

2014-4-25 15:16 回复 (1)

klous_6975

楼主你好！在消息处理函数里面为什么要加个锁呢？请不吝赐教！谢谢！

2014-7-17 14:19 回复 (0)

发评论

☐ 分享到微博 ☐ 匿名评论

验证码： 请点击后输入验证码 收听验证码

发评论

以上网友发言只代表其个人观点，不代表新浪网的观点或立场。

< 前一篇 QCompleter自动补全 后一篇 > Qt之日志输出窗口