# **Qt moveToThread 使用**

# **前置条件**

子线程对象代码结构：构造的时候会创建一个子线程，并将当前对象移动到这个线程中，此动作完成后，后续的槽将会在子线程中执行，此处传入了一个对象名用于后续调试；

#include <QDebug>  
#include <QThread>  
  
class TestThread : public QObject  
{  
 Q\_OBJECT  
  
public:  
 TestThread(QString objName, QObject \*parent = nullptr);  
 ~TestThread();  
signals:  
 void sgTestThreadEvent();  
public slots:  
 void slotDoTestEvent();  
private:  
 QString m\_thradName{ "" };  
 QThread\* m\_thread{ nullptr };  
};  
  
  
TestThread::TestThread(QString objName, QObject \*parent)  
 :m\_thradName(objName), QObject(parent)  
{  
 m\_thread = new QThread();  
 this->moveToThread(m\_thread);  
 connect(this, &TestThread::sgTestThreadEvent, this, &TestThread::slotDoTestEvent);  
 m\_thread->start();  
}  
  
TestThread::~TestThread()  
{}  
  
void TestThread::slotDoTestEvent()  
{  
 qDebug() << " objName: " << m\_thradName << \_\_FUNCTION\_\_;  
}

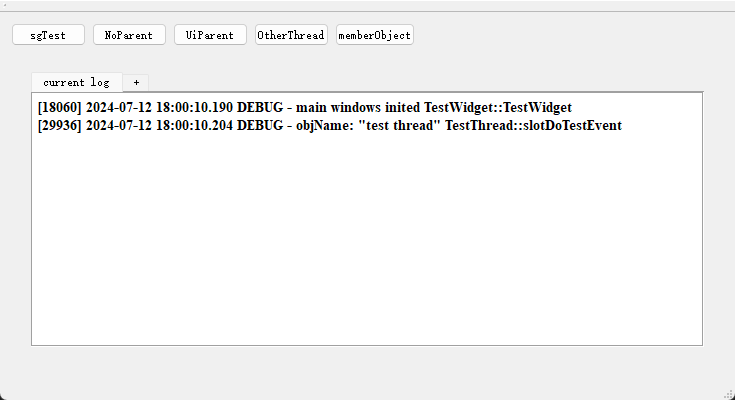
# **测试逻辑**

下面测试四种实现：

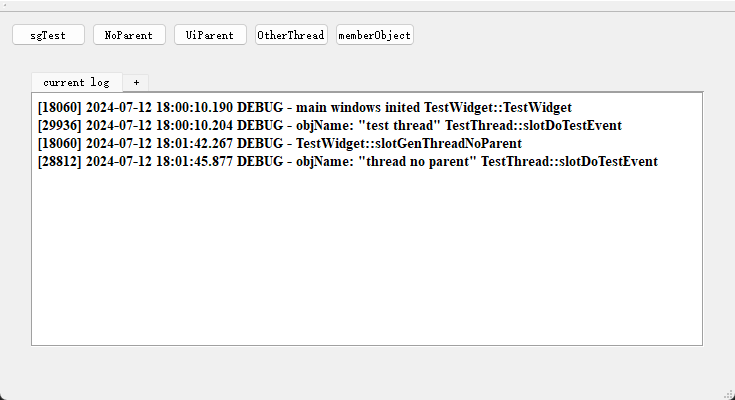
1. 父对象为 nullptr，即空指针的情况
2. 父对象为 ui 对象指针
3. 父对象为普通成员 QObject
4. 父对象为其他的线程对象

测试程序启动的时候将会输出一条主线程日志，以及创建一个测试线程对象；

下面为例：主线程 [ 18060 ], 测试线程对象 [ 29936 ]



## **2.1**父对象为 nullptr



主线程id为 18060 Line1

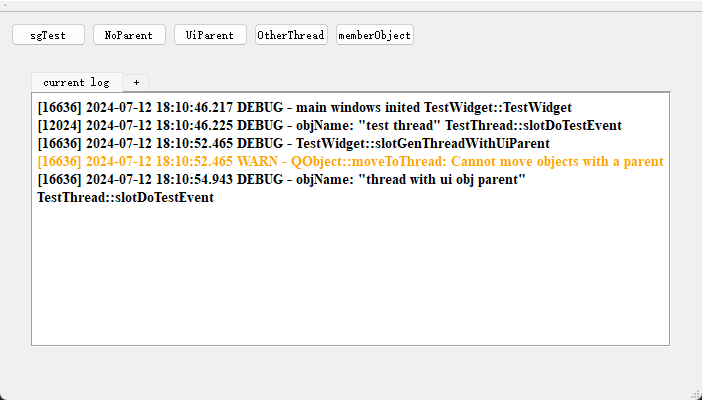
测试进程对象 id 为 29936 Line2

点击 NoParent 按钮创建子线程对象，无警告提示 Line3

点击 sgTest 通知子线程执行任务，子线程 id 为 28812 子线程创建成功 Line4

void TestWidget::slotGenThreadNoParent()  
{  
 if (m\_threadNoParent == nullptr)  
 {  
 qDebug() << \_\_FUNCTION\_\_;  
 // 创建子线程  
 m\_threadNoParent = new TestThread("thread no parent");  
 // 触发子线程任务  
 connect(this, &TestWidget::sgTestSignal, m\_threadNoParent,   
 &TestThread::sgTestThreadEvent);   
 }  
}

## **2.2** 父对象为 ui 对象指针



主线程id为 16636 Line1

测试进程对象 id 为 12024 Line2

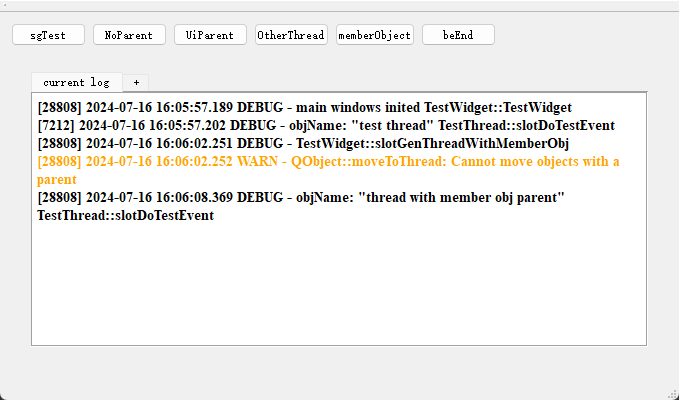
点击 UiParent 按钮创建子线程对象，系统给出警告 Line3

**QObject::moveToThread: Cannot move objects with a parent**

点击 sgTest 通知子线程执行任务，子线程 id 为 16636 子线程创建失败线程任务还是在主线程执行任务 Line4

void TestWidget::slotGenThreadWithUiParent()  
{  
 if(m\_threadWithUiParent == nullptr)  
 {  
 qDebug() << \_\_FUNCTION\_\_;  
 m\_threadWithUiParent = new TestThread("thread with ui obj parent", this);  
 connect(this,&TestWidget::sgTestSignal, m\_threadWithUiParent, &TestThread::sgTestThreadEvent);  
 }  
}

## **2.3** 父对象为普通成员 QObject



主线程id为 28808 Line1

测试进程对象 id 为 7212 Line2

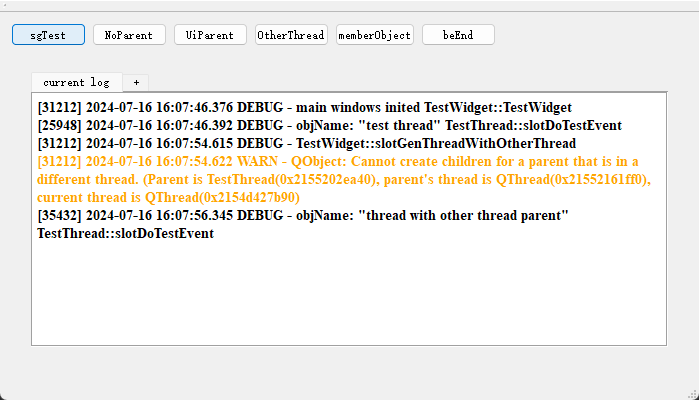
点击 UiParent 按钮创建子线程对象，系统给出警告 Line3

**QObject::moveToThread: Cannot move objects with a parent**

点击 sgTest 通知子线程执行任务，子线程 id 为 28808 子线程创建失败线程任务还是在主线程执行任务 Line4

void TestWidget::slotGenThreadWithMemberObj()  
{  
 if (m\_threadWithMemberObj == nullptr)  
 {  
 m\_testObject = new QObject();  
 qDebug() << \_\_FUNCTION\_\_;  
 m\_threadWithMemberObj = new TestThread("thread with member obj parent", m\_testObject);  
 connect(this, &TestWidget::sgTestSignal, m\_threadWithMemberObj, &TestThread::sgTestThreadEvent);  
 }  
}

## **2.4** 父对象为其他的线程对象



主线程id为 31212 Line1

测试线程 id 为 25948 Line2

点击 UiParent 按钮创建子线程对象，系统给出警告 Line3

**QObject: Cannot create children for a parent that is in a different thread. (Parent is TestThread(0x2155202ea40), parent's thread is QThread(0x21552161ff0), current thread is QThread(0x2154d427b90)**

**这个警告表示，当前的动作正在尝试将一个活动线程作为父对象指定给新建的目标线程**

点击 sgTest 通知子线程执行任务，会发现出现了一个新的线程号 35432，表示当前操作在一个新线程中启动，但是运行在一个完全不同的线程之中，和指定的父对象并非同一线程 Line4

void TestWidget::slotGenThreadWithOtherThread()  
{  
 if (m\_threadWithOtherThrad == nullptr)  
 {  
 qDebug() << \_\_FUNCTION\_\_;  
 m\_threadWithOtherThrad = new TestThread("thread with other thread parent", m\_testThread);  
 connect(this, &TestWidget::sgTestSignal, m\_threadWithOtherThrad, &TestThread::sgTestThreadEvent);  
 }  
}

# **结论**

如果一个逻辑模块希望使用moveToThread进入子线程，应该 "不指定明确的父对象" 或者 "设置父对象为 nullptr" ，其他操作将导致线程创建失败或者系统警告；