

Linux程序开发：QT的内部进程通信（2）PDF转换可能丢失图片或格式，建议阅读原文

[https://www.100test.com/kao\\_ti2020/144/2021\\_2022\\_Linux\\_E7\\_A8\\_8B\\_E5\\_BA\\_c103\\_144883.htm](https://www.100test.com/kao_ti2020/144/2021_2022_Linux_E7_A8_8B_E5_BA_c103_144883.htm) 在Qtopia中，所有的channels名都以"QPE/"开始，而messagename则是一个函数的标识符。在接收消息时，通常只需要利用在应用程序中预先定义好的QPE/Application/{appname}管道，当然，也可以根据需要自己定义管道，并将其与一个slot函数相关联：`myChannel = new QCopChannel("QPE/FooBar", this).connect(myChannel, SIGNAL(received(const QString amp.)), this, SLOT(fooBarMessage(const QString amp.)))`。下面将具体的通信过程举例如下，在需要接收消息的类（如Window1）中定义管道：`QCopChannel *doChannel = new QCopChannel("QPE/Do", this).connect(doChannel, SIGNAL(received(const QString amp.)), this, SLOT(doMessage(const QString amp.)))`。同时，需要在该类中定义相应的消息处理函数doMessage，`void Window1::doMessage(const QString amp.args) { QDataStream stream(args, IO_ReadOnly).if(msg == "Message1(QString)") { QString text. stream >> text. button->setText(text). } else if(msg == "Message2()") { close(). } }` 其中的Message1(QString)和Message2(QString)都是用户自己定义的消息，该函数中分别对这些消息进行了相应的处理。在该例中当收到带有参数的Message1消息时，将该字符串参数stream显示在按钮button上；当收到Message2消息时，将执行关闭Window1窗口的动作，当然用户可以根据需要自行编写相应的处理过程。另一方

面，在类Class2中需要发出消息的函数function中利用QCopEnvelope发送消息：`void Class2::function() { QCopEnvelope e("QPE/Do", "Message1(QString)"). e` 这里发出了Message1消息，并将需要携带的参数param发送到管道中。通过这样的过程，用户可以很方便地实现不同对象、不同进程之间通信过程，而且可以根据需要在通信过程中任意传递参数。100Test 下载频道开通，各类考试题目直接下载。详细请访问 [www.100test.com](http://www.100test.com)