<http://blog.csdn.net/liji_digital/article/details/52891213>

QT的 qdrag 和 qdrop功能在qt的"Puzzle"示例中有介绍。但是那个介绍是以QListWidget为基础的。本文以QWidget为基础，介绍如何实现拖放。拖放可以用QString作为信息载体，也可以使用html等类型。本文采用QByteArray。QByteArray允许程序员使用自定义的类型来作为信息载体。本文定义的类型是ST\_INFO。

头文件：

**[cpp]** [view plain](http://blog.csdn.net/liji_digital/article/details/52891213) [copy](http://blog.csdn.net/liji_digital/article/details/52891213)

1. #ifndef MAINWINDOW\_H
2. #define MAINWINDOW\_H
4. #include <QMainWindow>
5. #include <QDrag>
6. #include <QDropEvent>
7. #include <QMouseEvent>
8. #include <QLineEdit>
9. #include <QResizeEvent>

12. #pragma pack(1)
13. **struct** ST\_INFO
14. {
15. **bool** b;
16. **int**  i;
17. };
18. #pragma pack()
20. **class** MainWindow : **public** QMainWindow
21. {
22. Q\_OBJECT
24. **public**:
25. MainWindow(QWidget \*parent = 0);
26. ~MainWindow();
28. QPointF         m\_pntPress;
29. **bool**            m\_bMimeAdded;
30. QLineEdit     \* m\_pText1;
31. QLineEdit     \* m\_pText2;
32. QLineEdit     \* m\_pText3;
34. **protected**:
35. **void**    mousePressEvent(QMouseEvent \*);
36. **void**    mouseMoveEvent(QMouseEvent \*);
37. **void**    dropEvent(QDropEvent \*);
38. **void**    dragEnterEvent(QDragEnterEvent \*);
39. **void**    dragMoveEvent(QDragMoveEvent \*);
40. **void**    resizeEvent(QResizeEvent \*);
41. };
43. #endif // MAINWINDOW\_H

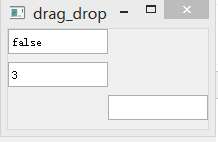
cpp文件：

**[cpp]** [view plain](http://blog.csdn.net/liji_digital/article/details/52891213) [copy](http://blog.csdn.net/liji_digital/article/details/52891213)

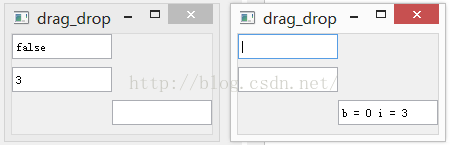
1. #include "mainwindow.h"
2. #include <QApplication>
3. #include <QMimeData>
4. #include <QDebug>
5. #include <QTextStream>
7. MainWindow::MainWindow(QWidget \*parent)
8. : QMainWindow(parent)
9. {
10. setAcceptDrops(**true**);
11. m\_pText1 = **new** QLineEdit(**this**);
12. m\_pText2 = **new** QLineEdit(**this**);
13. m\_pText3 = **new** QLineEdit(**this**);
14. }
16. MainWindow::~MainWindow()
17. {
18. **delete** m\_pText1;
19. **delete** m\_pText2;
20. **delete** m\_pText3;
21. }
23. **void** MainWindow::mouseMoveEvent(QMouseEvent \*event)
24. {
25. **if** ((QLineF(event->screenPos(), m\_pntPress)
26. .length() < QApplication::startDragDistance()) || m\_bMimeAdded)
27. {
28. **return**;
29. }
31. m\_bMimeAdded = **true**;
32. QDrag \*drag = **new** QDrag(**this**);
33. QMimeData \*mime = **new** QMimeData;
34. drag->setMimeData(mime);
36. QImage image("E:\\lena.jpg");
38. drag->setPixmap(QPixmap::fromImage(image).scaled(30, 40));
39. drag->setHotSpot(QPoint(15, 30));
41. QByteArray qba;
43. QString qstrTemp1 = m\_pText1->text();
44. QString qstrTemp2 = m\_pText2->text();
45. **bool** bTemp;
46. **if**(qstrTemp1.contains("true"))
47. {
48. bTemp = **true**;
49. }
50. **else**
51. {
52. bTemp = **false**;
53. }
54. ST\_INFO stTemp = {bTemp, qstrTemp2.toInt()};
55. qba.resize(**sizeof**(stTemp));
56. memcpy(qba.data(), &stTemp, **sizeof**(stTemp));
57. mime->setData(tr("INFO"), qba);

60. drag->exec();
61. }
63. **void** MainWindow::mousePressEvent(QMouseEvent \*e)
64. {
65. m\_pntPress = e->screenPos();
66. m\_bMimeAdded = **false**;
67. }
69. **void** MainWindow::dropEvent(QDropEvent \*event)
70. {
71. MainWindow \*source = qobject\_cast<MainWindow \*>(event->source());
72. **if**(source != **this**)
73. {
74. **if** (event->mimeData()->hasFormat("INFO"))
75. {
76. QByteArray pieceData = event->mimeData()->data("INFO");
78. ST\_INFO stTemp;
80. memcpy(&stTemp, pieceData.data(), **sizeof**(stTemp));
82. QString qstrTemp;
83. QTextStream stream(&qstrTemp, QIODevice::WriteOnly);
84. stream<<"b = "<<stTemp.b<<" i = "<<stTemp.i;
85. m\_pText3->setText(qstrTemp);
86. event->setDropAction(Qt::MoveAction);
87. event->accept();
88. }
89. **else**
90. {
91. event->ignore();
92. }
93. }
94. **else**
95. {
96. event->ignore();
97. }
98. }
100. **void** MainWindow::dragEnterEvent(QDragEnterEvent \*event)
101. {
102. **if** (event->mimeData()->hasFormat("INFO"))
103. event->accept();
104. **else**
105. event->ignore();
106. }
108. **void** MainWindow::dragMoveEvent(QDragMoveEvent \*event)
109. {
110. **if** (event->mimeData()->hasFormat("INFO")) {
111. event->setDropAction(Qt::MoveAction);
112. event->accept();
113. } **else** {
114. event->ignore();
115. }
116. }
118. **void** MainWindow::resizeEvent(QResizeEvent \*e)
119. {
120. m\_pText1->setGeometry(0,0,width()/2, height()/4);
121. m\_pText2->setGeometry(0, height()/3, width()/2, height()/4);
122. m\_pText3->setGeometry(width()/2, height()\*2/3, width()/2, height()/4);
123. }

启动两个一模一样的应用。在第一个窗口中输入如下：



然后在第一个窗口中按下鼠标，拖向第二个窗口，松开：



可见，第一个窗口的信息被正确传入了第二个窗口。