# Qt 之 QSystemTrayIcon

# 简述

QSystemTrayIcon类为应用程序在系统托盘中提供一个图标。

现代操作系统通常在桌面上提供一个特殊的区域，称为系统托盘或通知区域，长时间运行的应用程序可以显示图标和短消息。

# 内容

## 详细描述

要检查系统托盘是否存在在用户的桌面上，调用QSystemTrayIcon::isSystemTrayAvailable()静态函数。

要添加系统托盘项，首先，需要创建一个QSystemTrayIcon对象，并调用setContextMenu()为图标提供上下文菜单，然后，调用show()使其在系统托盘中可见。状态通知消息（“气球消息”）可以在任何时候使用showMessage()来进行显示。

当用户激活托盘图标时，activated()信号会被发射。

只有在X11上时，当发出一个tooltip请求时，QSystemTrayIcon会接收一个QEvent::ToolTip类型的QHelpEvent事件。此外，QSystemTrayIcon会接收QEvent::Wheel类型的滚轮事件。这些都是不支持任何其它平台。

## 公有类型

* 枚举QSystemTrayIcon::ActivationReason：

此枚举描述了系统托盘被激活的原因。

| **常量** | **值** | **描述** |
| --- | --- | --- |
| QSystemTrayIcon::Unknown | 0 | 未知原因 |
| QSystemTrayIcon::Context | 1 | 系统托盘的上下文菜单请求 |
| QSystemTrayIcon::DoubleClick | 2 | 双击系统托盘 |
| QSystemTrayIcon::Trigger | 3 | 单击系统托盘 |
| QSystemTrayIcon::MiddleClick | 4 | 鼠标中键点击系统托盘 |

* 枚举QSystemTrayIcon::MessageIcon：

此枚举描述了显示气球消息时所显示的图标。

| **常量** | **值** | **描述** |
| --- | --- | --- |
| QSystemTrayIcon::NoIcon | 0 | 无图标显示 |
| QSystemTrayIcon::Information | 1 | 一个信息图标显示 |
| QSystemTrayIcon::Warning | 2 | 一个标准的警告图标显示 |
| QSystemTrayIcon::Critical | 3 | 一个严重的警告图标显示 |

## 共有函数

* QMenu \* contextMenu() const   
  返回系统托盘的当前上下文菜单。
* void setContextMenu(QMenu \* menu)   
  设置指定菜单为系统托盘的上下文菜单。

当用户通过点击鼠标请求系统托盘的上下文菜单时，菜单会弹出。

在OS X中，一般转换为一个NSMenu，所以aboutToHide()信号不会发出。

**注意：**系统托盘菜单并不对菜单有所有权，必须确保在恰当的时候删除菜单，例如：创造一个具有合适父对象的菜单。

* QRect QSystemTrayIcon::geometry() const   
  返回系统托盘图标在屏幕上的几何坐标。
* QIcon icon() const
* void setIcon(const QIcon & icon)

icon : QIcon   
这个属性保存了系统托盘的图标。

在Windows中，系统任务栏图标的大小是16×16；X11中，首选大小为22x22。必要时该图标将被调整到合适大小。

* void setToolTip(const QString & tip)
* QString toolTip() const

toolTip : QString   
这个属性保存了系统托盘的提示信息。

在一些系统中，tooltip的长度是有限的，在必要时tooltip将被截断。

* bool isVisible() const   
  返回系统托盘是否可见。

## 公有槽函数

* void hide()   
  隐藏系统托盘。
* void setVisible(bool visible)   
  设置系统托盘是否可见。

设置为true或调用show()使系统托盘图标可见；设置为false或调用hide()隐藏它。

* void show()   
  显示系统托盘。
* void showMessage(const QString & title, const QString & message, QSystemTrayIcon::MessageIcon icon = QSystemTrayIcon::Information, int millisecondsTimeoutHint = 10000)   
  显示一个气球消息，使用所给出的标题、消息、图标和指定的时间，标题和消息必须是纯文本字符串。

消息可以被用户点击，当用户点击时发出messageClicked()信号 。

## 信号

* void activated(QSystemTrayIcon::ActivationReason reason)   
  当用户激活系统托盘图标，这个信号被发射。reason指定激活的原因， QSystemTrayIcon::ActivationReason列举了各种原因。
* void messageClicked()   
  当使用showMessage()显示的消息被用户点击时，此信号被发射。

目前，这个信号不会在OS X中发射。

## 静态共有函数

* bool isSystemTrayAvailable() [static]   
  如果系统托盘可用，返回true；否则，返回false。

如果系统盘是当前不可用，但以后变为可用，若QSystemTrayIcon可见，它就会自动在系统托盘中添加条目。

* bool supportsMessages() [static]   
  如果系统托盘支持气球消息，则返回true；否则，返回false。

# 示例

## 效果

 这里写图片描述 

## 源码

首先，自定义系统托盘需要使用的上下文菜单TrayMenu，其继承自QMenu，并通过addAction()为其添加需要的菜单项。

#include <QMenu>

class TrayMenu : public QMenu

{

Q\_OBJECT

public:

explicit TrayMenu(QWidget \*parent = 0);

~TrayMenu();

signals:

*// 自定义信号*

void showWindow();

private:

// 添加菜单项

void initActions();

private:

QAction \*m\_pShowAction;

QAction \*m\_pSettingAction;

QAction \*m\_pHomePageAction;

QAction \*m\_pHelpAction;

QAction \*m\_pAboutAction;

QAction \*m\_pUpgradeAction;

QAction \*m\_pQuitAction;

};

TrayMenu::TrayMenu(QWidget \*parent)

: QMenu(parent)

{

initActions();

}

TrayMenu::~TrayMenu()

{

}

*// 添加菜单项*

void TrayMenu::initActions()

{

*// 创建菜单项*

m\_pShowAction = new QAction(QIcon(":/icon/open"), QString::fromLocal8Bit("显示"), this);

m\_pSettingAction = new QAction(QIcon(":/icon/setting"), QString::fromLocal8Bit("设置"), this);

m\_pHomePageAction = new QAction(QIcon(":/icon/home"), QString::fromLocal8Bit("登录网页"), this);

m\_pHelpAction= new QAction(QIcon(":/icon/help"), QString::fromLocal8Bit("帮助"), this);

m\_pAboutAction = new QAction(QIcon(":/icon/about"), QString::fromLocal8Bit("关于"), this);

m\_pUpgradeAction = new QAction(QIcon(":/icon/upgrade"), QString::fromLocal8Bit("检查更新"), this);

m\_pQuitAction = new QAction(QIcon(":/icon/quit"), QString::fromLocal8Bit("退出"), this);

*// 添加菜单项*

addAction(m\_pShowAction);

addAction(m\_pSettingAction);

addAction(m\_pHomePageAction);

addSeparator();

addAction(m\_pHelpAction);

addAction(m\_pAboutAction);

addAction(m\_pUpgradeAction);

addSeparator();

addAction(m\_pQuitAction);

*// 连接信号-信号（仅列举连接显示窗口的信号）*

connect(m\_pShowAction, SIGNAL(triggered(bool)), this, SIGNAL(showWindow()));

}

然后，创建系统托盘QSystemTrayIcon，并通过setContextMenu()为其设置自定义的上下文菜单TrayMenu。

MainWindow::MainWindow(QWidget \*parent)

: CustomWindow(parent)

{

*// ...*

QSystemTrayIcon \*pSystemTray = new QSystemTrayIcon(this);

TrayMenu \*pTrayMenu = new TrayMenu(this);

*// 设置系统托盘的上下文菜单*

pSystemTray->setContextMenu(pTrayMenu);

*// 设置系统托盘提示信息、托盘图标*

pSystemTray->setToolTip(QString::fromLocal8Bit("我就是托盘"));

pSystemTray->setIcon(QIcon(":/icon/login"));

*// 连接信号槽*

connect(pTrayMenu, SIGNAL(showWindow()), this, SLOT(showWindow()));

connect(pSystemTray , SIGNAL(activated(QSystemTrayIcon::ActivationReason)), this, SLOT(onActivated(QSystemTrayIcon::ActivationReason)));

*// 显示系统托盘*

pSystemTray->show();

*// 显示系统托盘提示信息*

pSystemTray->showMessage(QString::fromLocal8Bit("托盘标题"), QString::fromLocal8Bit("托盘显示内容"));

}

*// 显示窗体*

void MainWindow::showWindow()

{

showNormal();

raise();

activateWindow();

}

*// 激活系统托盘*

void MainWindow::onActivated(QSystemTrayIcon::ActivationReason reason)

{

switch(reason)

{

*// 单击托盘显示窗口*

case QSystemTrayIcon::Trigger:

{

showNormal();

raise();

activateWindow();

break;

}

*// 双击*

case QSystemTrayIcon::DoubleClick:

{

*// ...*

break;

}

default:

break;

}

}