

User Interface Programming Qt

Jozef Mlích

*Department of Computer Graphics and Multimedia
Faculty of Information Technology, Brno University of Technology
Božetěchova 2, 612 66 Brno, Czech Republic
<http://www.fit.vutbr.cz/~imlich>
imlich@fit.vutbr.cz*





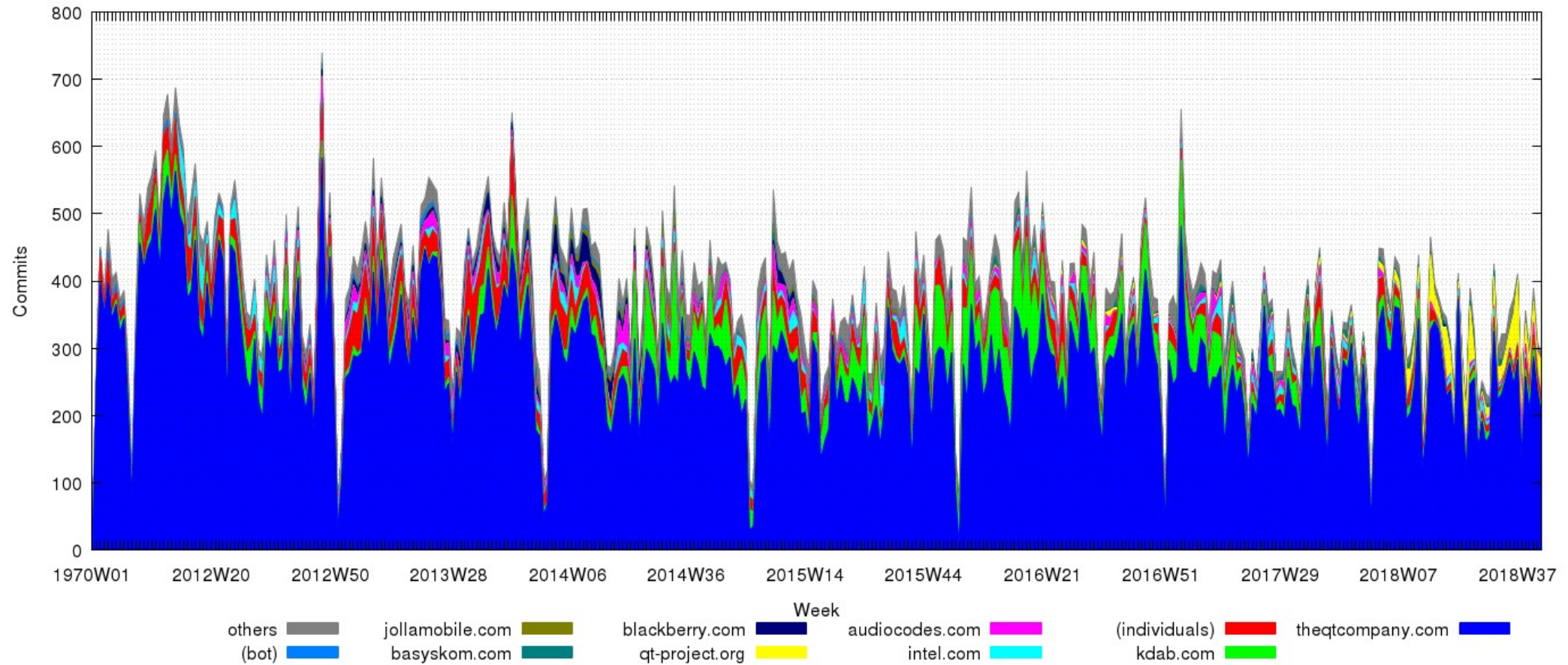
**Code less.
Create more.
Deploy everywhere.**

- Motivation, Features, History
- Architecture of Qt
- qmake and other tools
- QObject
 - Meta object model,
 - Events, Signals and slots,
 - Properties,
 - Memory management
- Widgets
- Resources
- Object positioning
- i18n

History

- 1991 – Haavard Nord and Eirik Chambe-Eng (Windows/Linux)
- 1994 – Trolltech
- 1996 – KDE
- 2001 – Mac Support
- 2005 – All platforms are under GPL
- 2008 – Nokia
- 2012 – qt-project.org, Digia

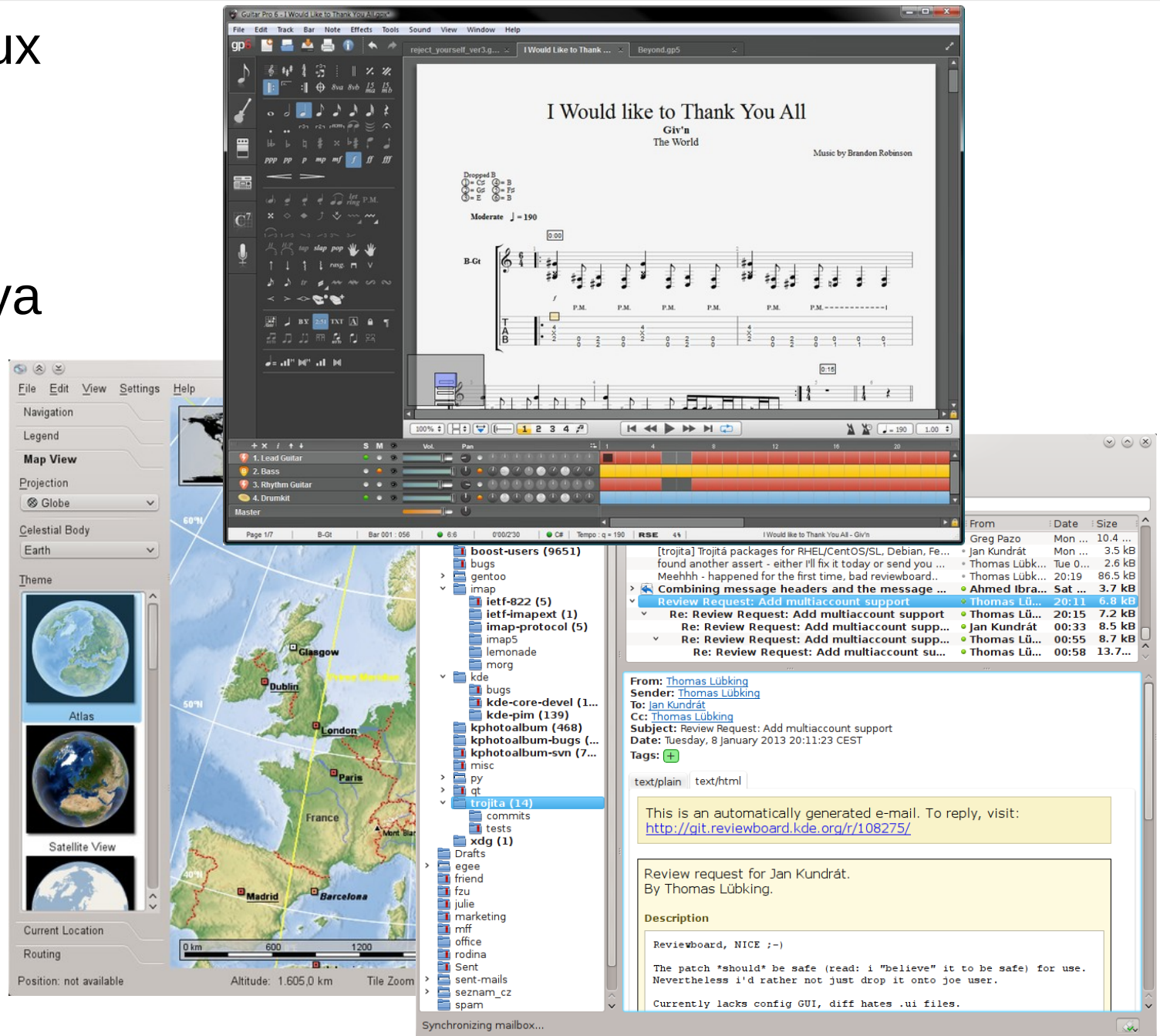
Who is making Qt?



Thiago Macieira, <http://www.macieira.org/blog/qt-stats/>

Who is using Qt?

- Skype for Linux
- Team Speak
- Google Earth
- Autodesk Maya
- Guitar Pro 6
- VLC player
- GNU Octave
- OpenSCAD



Features

- C++, Python (PySide2), (~~C#, Java — Jambi~~)
- API for everything (not only GUI)
- cross-platform
 - Windows
 - Mac
 - Linux
 - Embedded systems (via “frame buffer”, Qt on Pi)
 - Mobile devices (Symbian, Maemo, MeeGo, Jolla, Blackberry, Android, iOS)
- Licence
 - GPL, LGPL, Commercial

CORE

multithreading

databases

scripting

networking

XML

DBus

GUI

declarative UI

multimedia

2d canvas

QtWebKit

OpenGL

Qt Mobility

unit testing

Qt Creator

Environment for Qt Apps Debugging

~/ .bashrc

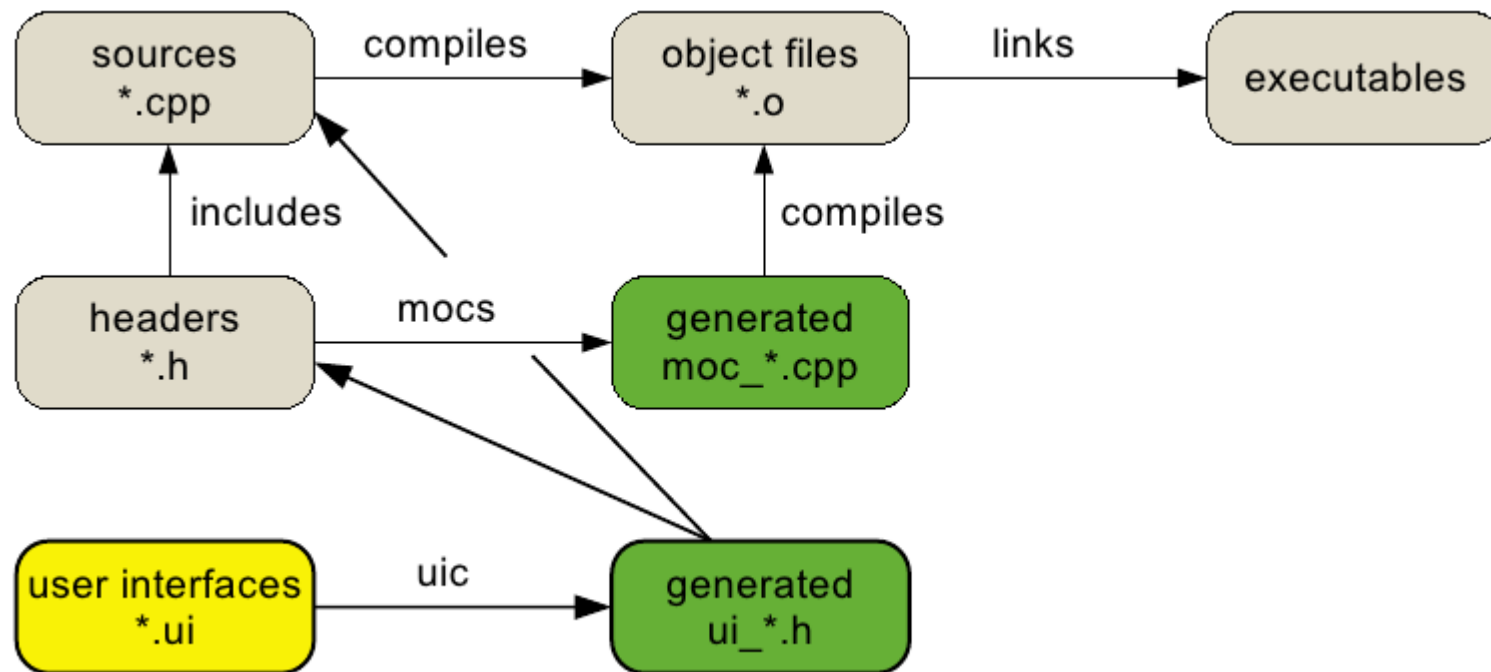
```
# avoid console.log() supression
export "QT_LOGGING_RULES=*=true
qt.*=false"
```

```
# integration with gnome troubles
# export QT_QPA_PLATFORMTHEME=gtk2
```

```
# nice git prompt
source /usr/share/git-core/contrib/completion/
git-prompt.sh
export PS1='[\u@\h \W$(declare -F __git_ps1
&>/dev/null && __git_ps1 " (\[\e[1;31m\]%s\[\e[0m\])")\$ '
```

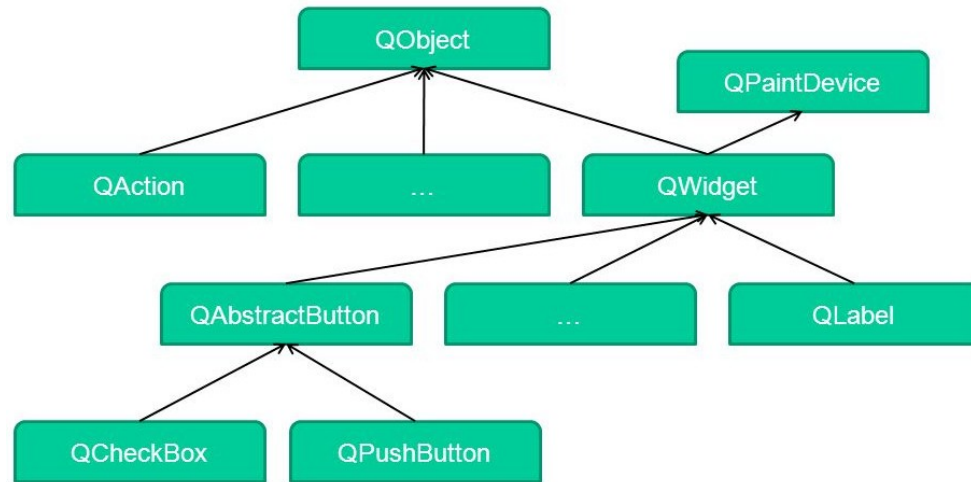
Compilation and tools

- qmake, (moc, uic)
- (cmake, automoc)
- Qt Creator, Qt Linguist, (Qt Designer), qmlscene (qmlviewer)



- Qt extends C++ - macro language, introspection, etc.
 - `foreach (int value, intList) { ... } // (i.e. Q_FOREACH)`
 - `QObject *o = new QPushButton;`
`o->metaObject()->className(); //returns "QPushButton"// introspection`
 - `connect(button, SIGNAL(clicked()), window, SLOT(close()));`
- It is still C++
- QObject
 - Events
 - Signals and slots
 - "Properties"
 - Memory management

```
class MyClass : public QObject
{
    Q_OBJECT
public:
    MyClass(QObject *parent = 0);
    ~MyClass();
signals:
    void mySignal(int);
public slots:
    void mySlot(int);
};
```

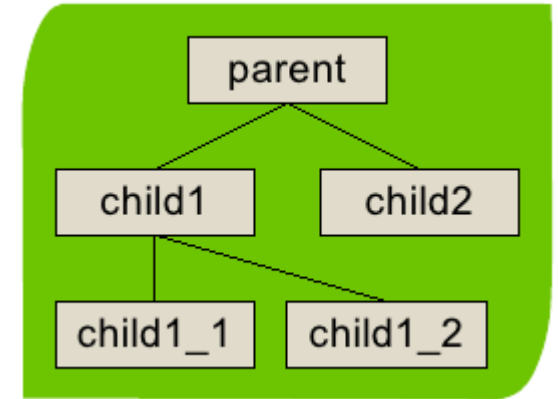


- QT += widgets
(vs. QML + Quick)
- Widget is basic graphical primitive in Qt
- Objects are in Tree Hierarchy
- Its behavior is given by its source code and by inheritance.
- (Almost) Every visual object is child `QWidget` in terms of inheritance

- Sample Code

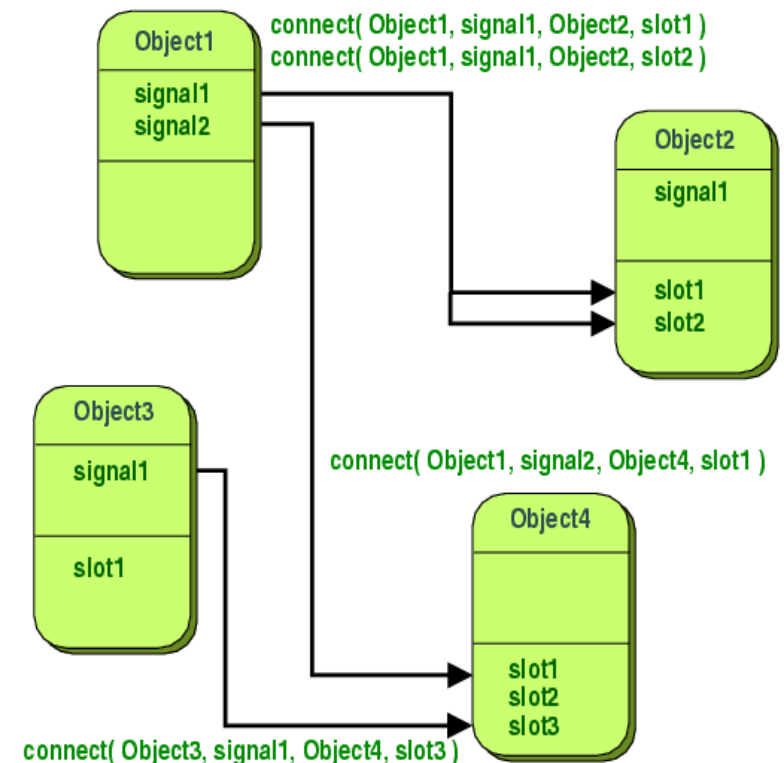
```
QObject *parent = new QObject();  
QObject *child1 = new QObject(parent);  
QObject *child2 = new QObject(parent);  
QObject *child1_1 = new QObject(child1);  
QObject *child1_2 = new QObject(child1);  
  
delete parent;
```

- This is used when implementing visual hierarchies.
- Heap memory is explicitly allocated and freed using new and delete
- Local variables are allocated on Stack



Signals and slots

- A “callback” (in other tool kits) is a pointer to a function that is called when an event occurs, any function can be assigned to a callback
 - No type-safety
 - Always works as a direct call
- Signals and Slots are more dynamic
 - A more generic mechanism
 - Easier to interconnect two existing classes
 - Less knowledge shared between involved classes

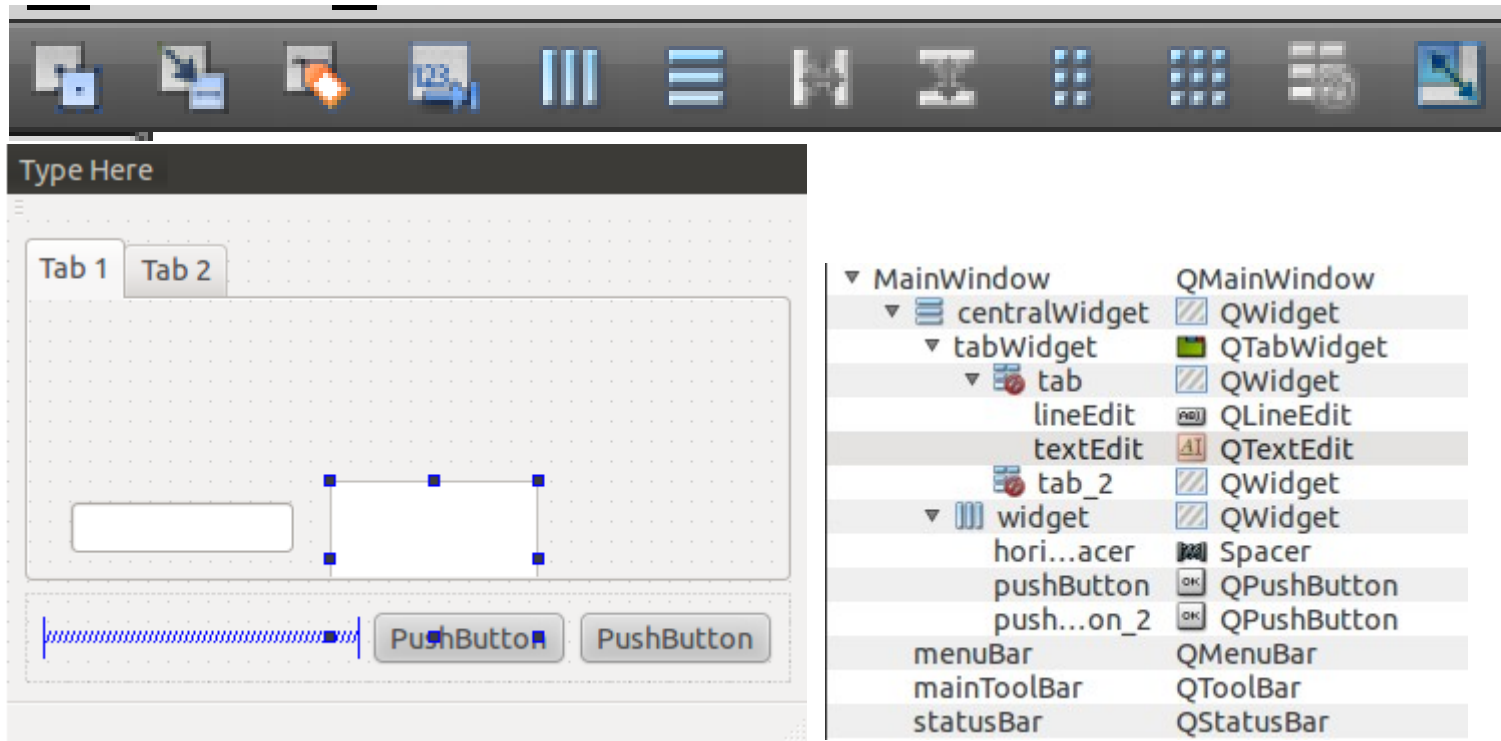
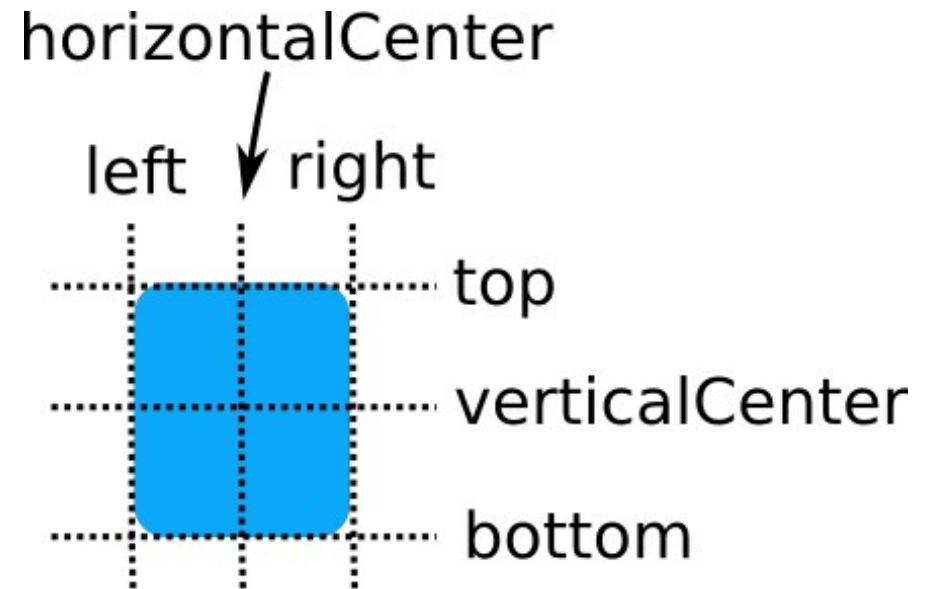


- Used for data embedded into the executable file
 - Images
 - Icons
 - Sounds
 - (QML Source code)

```
qUrl(QLatin1String("qrc:/camera-icon.png"));
```

Object positioning

- Layouts
- Absolute positions
- Anchors
- ...



- `QTranslator trans;`
`trans.load("hello_cs_CZ");`
`app.installTranslator(&trans);`
`new QLabel(tr("Password:"));`
`showMessage(tr("%n
message(s) saved", "", n);`
`//% "annotation %1 with int"`
`qsTrId("engineering-
english").arg(number)`
- `lupdate hello.pro`
- `linguist ./i18n/*.ts`
- `lrelease -idbased ./i18n/*.ts`

```
LANGUAGES = cs_CZ en_US

# var, prepend, append
defineReplace(prependAll) {
    for(a,$$1):result += $$2$$a$$$3
    return($$result)
}

unix: {
    LRELEASE = lrelease-qt5
}
win32: {
    LRELEASE = lrelease
}

TRANSLATIONS = $$prependAll(LANGUAGES,
    $$PWD/i18n/hello_*.ts)

updateqm.input = TRANSLATIONS
updateqm.output = $$OUT_PWD/$
    {QMAKE_FILE_BASE}.qm
updateqm.commands = $$LRELEASE -idbased -
    silent ${QMAKE_FILE_IN} -qm $
    {QMAKE_FILE_BASE}.qm
updateqm.CONFIG += no_link target_predeps
QMAKE_EXTRA_COMPILERS += updateqm

qmfiles.files = $$prependAll(LANGUAGES, $
    $OUT_PWD/viewer_*.qm)
qmfiles.path = $$PREFIX/share/${TARGET}/i18n
qmfiles.CONFIG += no_check_exist

INSTALLS += qmfiles
```

References

- <http://doc.qt.io/>
- <https://qmlbook.github.io/>
- IRC: #qt and #qt-qml at freenode
- Fitzek, Frank H. P., Tommi Mikkonen, and Tony Torp. Qt for Symbian. John Wiley and Sons, 2010.
- Molkentin, D: The Book of Qt 4: The Art of Building Qt Applications, No Starch Press, 2007, 440p, ISBN10: 1593271476, ISBN13: 978159327147

Questions ?