

A background image of a glowing lightbulb with a dark, textured base, set against a dark blue background with soft, out-of-focus light spots. The lightbulb is tilted and appears to be the source of the light.

Qt For Beginners Training

Part 2

© Integrated Computer Solutions, Inc.
© Integrated Computer Solutions, Inc.
All Rights Reserved
All Rights Reserved

© Integrated Computer Solutions, Inc.
All Rights Reserved

Qt's Classes

- Qt's 1,500 classes can be divided into two groups
 - Identity types
 - Value types
- Identity types
 - derive from QObject
 - extend C++ with many dynamic features using a meta-object system
 - cannot be copied as the copy constructor and assignment operator equal to delete
 - QWidget, QMainWindow, QApplication, QEventLoop, QThread, QFile, QTcpSocket
- Value types are standard C++ classes
 - QColor, QEvent, QDataStream, QMetaType
 - ~100 value types use copy-on-write pattern – implicitly shared
 - QString, QByteArray, QList, QVector, QHash, QCache, QDir, QPixmap, QImage, QBrush, QPen

Qt's C++ Object Model - QObject

- **QObject** is the heart of Qt's object model

- Include these features:

- Memory management
- Object properties
- Introspection
- Signals and Slots
- Event handling

- **QObject** has no visual representation

© Integrated Computer Solutions, Inc.
All Rights Reserved

Meet QVariant

- **QVariant**
 - Union for common Qt "value types" (copyable, assignable)
 - Supports implicit sharing (fast copying)
 - Supports user types
- A generic data object

© Integrated Computer Solutions, Inc.
All Rights Reserved

QVariant

- For QtCore types

```
QVariant variant(42);  
int value = variant.toInt(); // read back as integer  
QString text = variant.toString(); // read back as string  
QDebug() << variant.typeName(); // int
```

© Integrated Computer Solutions, Inc.

- For non-core and custom types:

```
QVariant variant = QVariant::fromValue(QColor(Qt::red));  
QColor color = variant.value<QColor>(); // read back  
QDebug() << variant.typeName(); // "QColor"
```

Making Your Custom Classes QVariants

```
#include <QMetaType>
```

```
class Contact
```

```
{
```

```
public:
```

```
    void setName(const QString & name);
```

```
    QString name() const;
```

```
    ...
```

```
};
```

```
Q_DECLARE_METATYPE(Contact);
```

- Type must support default construction, copy and assignment.
- **Q_DECLARE_METATYPE** should be after **class** definition in header file.

qRegisterMetaType

- **Q_DECLARE_METATYPE** makes the type known to **QVariant** and template based functions
- **qRegisterMetaType** will register the type

© Integrated Computer Solutions, Inc.

```
const int typeId = qRegisterMetaType<Contact>();
```

- Type can be used with property system and as signal parameters in queued connections

Properties

- **Q_PROPERTY** is a macro:

```
Q_PROPERTY (type name READ getFunction  
            [WRITE setFunction] [RESET resetFunction]  
            [NOTIFY notifySignal])
```

© Integrated Computer Solutions, Inc.

All Rights Reserved

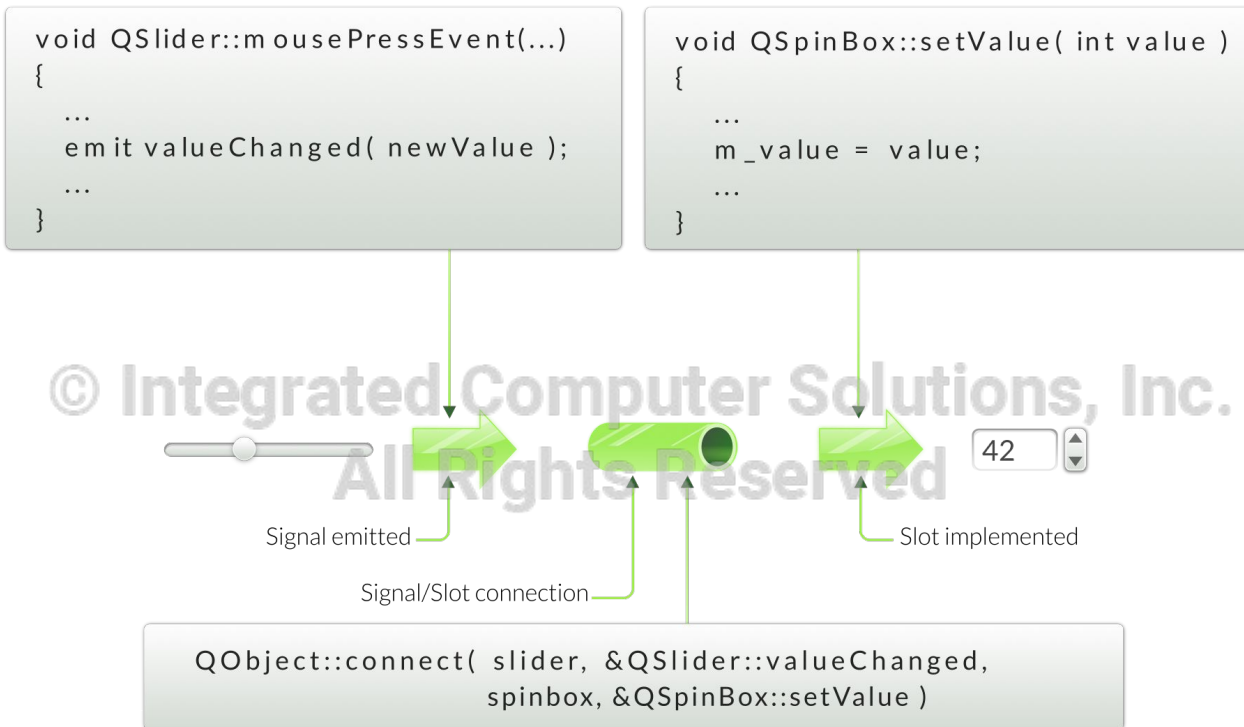
- Property access methods:

```
QVariant property(const char* name) const;
```

```
void setProperty(const char* name, const QVariant& value);
```

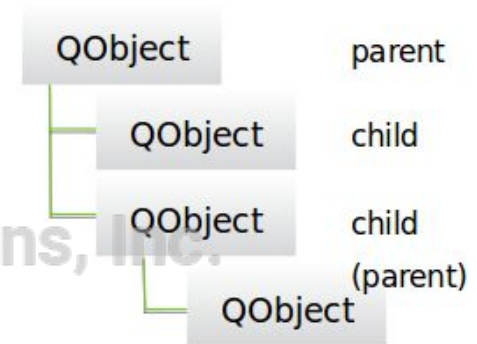
- Note:
 - **Q_OBJECT** macro is required for **Q_PROPERTY** to work

Connecting Signals to Slots



QObject Tree

- QObject objects organize themselves in object trees
 - Based on parent-child relationship
- `QObject(QObject *parent = Q_NULLPTR)`
 - Parent adds object to list of children
 - `const QObjectList &children();`
 - `QList<QWidget> findChildren();`
 - Parent owns children
- Construction/Destruction
 - Tree can be constructed in any order
 - Tree can be destroyed in any order
 - if object has parent: object first removed from parent
 - if object has children: deletes each child first
 - No object is deleted twice
- *Note: Parent-child relationship is NOT inheritance*



Further Reading

- The Qt Property System

<http://doc.qt.io/qt-5/properties.html>

- How Signals and Slots Work

<https://woboq.com/blog/how-qt-signals-slots-work.html>

In the Next Sessions

- QML
- QWidgets
- Model/View

© Integrated Computer Solutions, Inc.
All Rights Reserved