The Many Ways to Unite QML and C++

Way 0 - QtQuick UI

- Powerful
- Simple
- Common

```
import QtQuick.Particles 2.0
ParticleSystem {
    anchors.fill: parent
    ImageParticle {
        source: "qrc:///particleresources/star.png"
        alpha: 0.8
        colorVariation: 1.0
        z:10000
    }

    Emitter {
        id: burstEmitter
        anchors.centerIn: parent
        emitRate: 1000
        lifeSpan: 2000
        enabled: false
        velocity: AngleDirection {
            magnitude: 64;
            angleVariation: 360
        }
        size: 24
        sizeVariation: 8
    }
}
```

Way 1 - JSON++

- Can you spot all the errors?
- Do you get any help?
- Impact at parse or run time?
- It is human readable...
- But is it human writable?

```
{
  'alpha' : 1,
  'beta' : 2,
  'gama' : 3,
  'numberOfThings: 'four',
}
```

Way 1 - JSON++

- Comments
- Default values
- Typed data
- Parse error on misspelling

```
import QtQml 2.2
QtObject {
    property int alpha
    property int beta
    property int gamma
    //Can use comments, which is nice
    property int numberOfThings: 4
}

import "DataDefinition"
Datum {
  alpha: 1
  beta: 2
  gama: 3
  numberOfThings: "four"
}
```

Way 1 - JSON++

- Easy Qt/C++ for JSON
- For reading it
- Error handling very manual

```
QFile file("data.json");
if (!file.open(QIODevice::ReadOnly | QIODevice::Text))
    return handleError("Cannot open file");
QJsonParseError err;
QJsonDocument jd =
        QJsonDocument::fromJson(file.readAll(), &err);
if (err.error != QJsonParseError::NoError)
    return handleError(err.errorString());
if (!jd.isObject())
    return handleError("File valid, but not object");
QJsonValue alpha = jd.object().value("alpha");
if (alpha.isUndefined())
    return QLatin1String("defaultValue");
if (!alpha.isString())
    return handleError("Alpha value is not a string");
return alpha.toString();
```

Way 1 - JSON++

- Easy Qt/C++ for QML
- Error handling done for you
- Errors as array as well
- Also turns data into QObjects

```
QQmlEngine engine;
QQmlComponent c(&engine, "data.qml");
QObject* qmlObject = c.create();
if (!qmlObject)
    return handleError(c.errorString());
return qmlObject->property("foo").toString();
```

Way 2 - Impatient Scripting

- Unreal QtScript replacement
- JS functions
- Pass in as args or context

```
import QtQml 2.0
QtObject {
    function getRand() {
        setRand(Math.floor(Math.random() * 200)
                + 100;
    }
    Timer {
        interval: 1000
        running: true
        repeat: true
        onTriggered: saveDoc();
    }
}
```

Way 2 - Impatient Scripting

- QML Engine does type conversion
- Network transparent
- C++ side is QObject<->QObject
- Plus context

```
QQmlEngine engine;
engine.rootContext()->setContextObject(enabler);
QUrl url("https://trusted.server.com/qml/custom.qml");
QQmlComponent c(&engine, url);
qmlObject = c.create();
if (!qmlObject)
    return handleError(c.errorString());
//Connection could be done in QML too!
QObject::connect(enabler, SIGNAL(newRand()),
        qmlObject, SLOT(getRand()));
```

Way 2 - Impatient Scripting

- Script can be directed
- Custom object harness
- Or something like QQSM

```
import QtQml.StateMachine 1.0
StateMachine {
    initialState: start
    FinalState {
        id: final
    }
    StateBase {
        id: start
        SignalTransition {
            targetState: middle
        }
    }
    StateBase {
        id: middle
        SignalTransition {
            targetState: final
        }
    }
}
```

Way 2 - Impatient Scripting

- No Sandbox
- Imports can load native code
- Can't (yet) disable QtQuick

```
import QtQuick.Window 2.1
import QtQuick 2.1

Window {
    width: 1920
    height: 1280
    visible: true
    Rectangle {
        color: "blue"
        anchors.fill: parent
        Text {
            color: "red"
            text: "I HAXXED UR MAHCINE"
            anchors.centerIn: parent
            font.pixelSize: 128
        }
    }
}
```

Way 3 - Enriched Data

- Combining the last two
- Data + Logic
- Are we programming yet?

```
import "DataDefinition"
Datum {
  alpha: 1
  beta: 2
  gama: 3
  numberOfThings: "four"
  function getRand() {
      setRand(Math.floor(Math.random() * 200)
              + 100;
  }
}
```

Way 3 - qbs

- qtgerrit:qt-labs/qbs.git
- Advanced Config Files
- Snippet from qtcreator.qbs

```
import qbs 1.0

Project {
    minimumQbsVersion: "1.3"
    property bool withAutotests: qbs.buildVariant === "deb
    [...]
    property string ide_bin_path: qbs.targetOS.contains("d
            ? ide_app_target + ".app/Contents/MacOS"
            : ide_app_path
    property bool testsEnabled: qbs.getEnv("TEST") || qbs.
    property stringList generalDefines: [
        "QT_CREATOR",
        'IDE_LIBRARY_BASENAME="' + libDirName + '"',
        "QT_NO_CAST_TO_ASCII",
        "QT_NO_CAST_FROM_ASCII"
    ].concat(testsEnabled ? ["WITH_TESTS"] : [])
    qbsSearchPaths: "qbs"
    references: [
        "src/src.qbs",
        "share/share.qbs",
        "share/qtcreator/translations/translations.qbs",
        "tests/tests.qbs"
    ]
}
```

Way 3 - KineticTD

- Unit data for a TD
- Parameters for managed instance
- + Full UI control

```
MovingUnit {
    id: innerContainer
    speed: 1
    hp: 1
    onHpChanged: if(hp<=0) innerContainer.kill();
    [...]
    width: 10
    height: 10
    Rectangle{
        id: rect2
        anchors.fill: parent
        radius: 4
        border.color: "black"
        color: "white"
    }
}
```

Way 3 - Dialogue Tree

- Complex Dialogue Tree
- Allows non-trivial branching
- Data + Logic

```
import "dialogue" as Dialogue
Dialogue.TestContent{
    id: story
    onCurrentNodeChanged: {
        if(currentNode.prechoiceText != "")
            historyText.append(currentNode.prechoiceText)
        choiceBox.model = currentNode.choices;
    }
}

ScrollView {
    id: dialogue1
    anchors.fill: parent
    anchors.bottomMargin: 300
    Text {
        id: historyText
        width: parent.width
        wrapMode: Text.WordWrap
        text: "Welcome to Space Pirates VS Socrates!"
        function append(newStuff) {
            historyText.text = historyText.text
                    + "\n" + newStuff
        }
    }
}
```

Way 3 - Dialogue Tree

Welcome to Space Pirates VS Socrates!

New Game
Save Game
Load Game
Exit Game
Quick Game
Game Options
Why not call it 'Options Game' and maintain the pattern? Why even have a separate options screen at all? You're already ruining the immersion and thus the entire game!