

# QML's many faces

Kevin Krammer  
FOSDEM 2013

# Declarative Language + Engine

```
import QtQuick 1.1
```

```
Item {
```

```
    TextInput {
```

```
        id: input
```

```
        text: "QtQuick"
```

```
    }
```

```
    Rectangle {
```

```
        width: input.width
```

```
        height: 5
```

```
    }
```

```
}
```

BB10/Cascades  
BlackBerry

# BB10/Cascades

```
import bb.cascades 1.0
```

```
Container {
```

```
    TextField {
```

```
        id: input
```

```
        text: "QtQuick"
```

```
    }
```

```
    Label {
```

```
        text: input.text
```

```
    }
```

```
}
```

# Declarative Widgets

## KDAB

# Declarative Widgets

```
import QtGui 1.0
```

```
Widget {
```

```
   LineEdit {
```

```
        id: input
```

```
        text: "QtQuick"
```

```
    }
```

```
    Label {
```

```
        text: input.text
```

```
    }
```

```
}
```

# QBS Qt Project

# QBS

```
import qbs.base 1.0
```

```
Application {  
    files: [ "main.cpp" ]  
  
    Depends {  
        name: "Qt"  
        submodules: [ "core", "declarative" ]  
    }  
}
```



# Declarative CMake

## KDAB

# Declarative CMake

```
import org.CMake 2.8
```

```
CMake {
```

```
  Target {
```

```
    id: program
```

```
    name: "helloworld"
```

```
    files: [ "main.cpp" ]
```

```
    Package {
```

```
      name: "Qt5Qml"
```

```
    }
```

```
  }
```

```
  Install {
```

```
    destination: "bin"
```

```
    operands: program
```

```
  }
```

```
}
```

Game Logic  
Just brainstorming

# Game Logic

```
import game 1.0
```

```
Level {
```

```
  Switch {
```

```
    id: switch1
```

```
  }
```

```
  Door {
```

```
    locked: switch1.toggled
```

```
  }
```

```
}
```

Packaging  
Just brainstorming

# Packaging

```
import package 1.0
```

```
Package {  
  name: "Qt Creator"  
  
  Depends {  
    Package {  
      name: "Qt5Libs"  
    }  
  }  
  
  Recommends {  
    Package {  
      name: "Qt5Documentation"  
    }  
  }  
}
```

Email Notification  
Just brainstorming

# Email Notification

```
import email 1.0
```

```
Email {  
  subject: "Reminder" + _event.title  
  to: _event.participants  
  bcc: _preferences.encrypted ? [ _event.organizer ] : []  
  
  MultiPart {  
    mime: "text/html"  
    content: _event.descriptionHTML  
  }  
  
  MultiPart {  
    mime: "text/plain"  
    content: _event.description  
  }  
}
```



# Summary

- Declarative Language + Engine
- Custom Component sets
- Builds tree of objects
- Dependency chains
- Runtime property updates

Feedback welcome! [krammer@kde.org](mailto:krammer@kde.org)