QtJambi

Java bindings for the powerful Qt framework

Qt

- Powerful framework for C++
- GUI, XML, Database, IO, Network, Threads...
- Crossplatfrom including embedded systems
- Double license - GPL & commertial
- Bindings for Java, Python,Perl, PHP ...
- QtScript - build extensible application using EcmaScript
- and much more...
- **But don't we have all these features in java already?**

GUI - the main reason to use QtJambi

- Native look and feel on all platforms
- Widgets (= GUI components): buttons, select boxes, tables...
- Subtype QWidget to make your own component
  - easy painting with QPainter (lines, circles, texts)
  - anti-aliasing
  - matrix transformations
  - alpha channel
  - embed other widgets
  - hardware acceleration with OpenGL
- UI editor (standalone or as Eclipse plugin)

Signals and slots

```
public class MyGUI {

  private QPushButton button;

  public MyGUI() {
    button = new QPushButton("Go!");
    button.clicked
      .connect(this,"doSomething()");
  }

  public void doSomething() {
    System.out.println("hello world");
  }
}
```

Signals and slots

Not the Java-way:

- the *clicked* signal is a public field
- slot *doSomething()* referenced as a string
- doSomething() can even be private
- refactoring problematic

But:

- fewer LOC
- passing of arguments possible
- safe alternative possible (much longer though)

"On a side note... Qt has been using strings for signal / slot connections for over a decade and in practice this has proven to not be a major hurdle. The reason for this is that connections are typically made during the init phase of objects and therefore always checked and fixed at an early stage."

--QtJambi FAQ

Other cool feaures

- Easy WebKit integration
- Native libraries - simple deployment as jar
- Deployment as Java WebStart app possible
- Easy internationalization
- Phonon integration - crossplatform multimedia playback
- Drag&drop
- Crossplatform system tray
- Style sheets

The downside

- Sometimes not "the Java way"
- Overlapping functionality with standard Java classes
  - QThread
  - QFile
  - ...
- Difficult debugging because of native libraries
- Documentation - C++ code snippets in examples
- Everything starts with Q
- Stability (in early versions)

So is it worth it?

Yes!

(given enough time)