

Small introduction to Qtopia

Holger Freyther

What is Qtopia

- Based on Qt
- A set of libraries for mobile devices
- A set of services (intents?)
- Libraries + Services = Framework?
- A set of Applications
- <http://trolltech.com/products/qtopia>

What is Qt

- Cross platform toolkit
- Code less, create more, a productivity tool
- Supports X11, Embedded, Windows, Windows CE, Mac
- High quality, developers are proud to work on it.
- <http://www.trolltech.com/products/qt>

History of Qtopia

- Started as demonstration for Qt/Embedded
- Was named QPE and hosted on qpe.sourceforge.net
- Was made a product for Sharp, PDA
- Was turned into a phone stack
- VoIP and other stuff added
- Shows the flexibility of the code, things are bit rushed though. Lower quality than Qt

The Qtopia libraries

- src/libraries inside the tree
- Includes the base libraries, communication, phone, multimedia, mail
- Extend Qt to have mobile/desktop specific libraries.

Qtopia Infrastructure

- QCOP for Interprocess Communication, predates D-Bus
- Loading of images and other resources
- Services
- Runtime services by 'qpe'

QCOP

- Concept of channels.
- System channel: Only one system channel
- Application channel: each running app has a channel
- Other channels
- You send a named message and a QByteArray
- marshalling/demarshalling done by hand
- <http://doc.trolltech.com/qtopia4.3/qtopiaipcenvelope.html>,
<http://doc.trolltech.com/qtopia4.3/qtopiaipcadaptor.html>

Qtopia Services

- You define a service. Check `services/*.service`
- You set yourself as provider of this service by touch `services/FooService/MyApp`
- With more than two providers you can pick your preferred application
- Example services: Dial, Show Mails, Shutdown
- `qcop service send Launcher execute(QString) foo`

The qpe process

- `src/server`
- Launches applications, passes on messages, starts the modem, provides the homescreen, softmenu and other things.
- Has builtin apps
- Window management to some degrees
- Soft Menu: `ContextLabel`, `setLabelText` `QCOP` message to send the text.
- Is doing a lot, too much?

Approach to Qtopia on X11

Goals and Requirements

- Integrate with Illume
- Make it work
- Be able to do phonecalls
- Should be done as fast as possible
- Should end up upstream at Trolltech
- Should be able to upgrade to the new versions of Qtopia, once they release them.

Approach to Qtopia on X11

Coding Style

- Coding principal: Ugly code does not work
- Code with too many `#ifdefs`, not fitting on my screen, mixed naming of variables, code duplication is ugly.
- Clean up before you add your junk, think of where it should belong.
- Don't destroy the existing structure (e.g. adding globals), refine it, make the code better.

Approach to Qtopia on X11

Version Control Systems

- One big commit once it is done?
- Many small ones on the way?
- One commit per feature/bug/issue/improvement
- Makes it easy to review
- Makes it easy to remember what you did
- Makes it easy to merge new versions
- With git you do not spam the mailinglist when doing many small commits.

Approach to Qtopia on X11

Keeping track of issues

- Use the Bugzilla or Wiki to track your issues
- Describe them in a way others could understand them and you remember what you meant
- It is fine to not fix all issues at once, not having the best solution right now, keep track of it, but change your code once you can do the best solution.
- wiki.openmoko.org/wiki/QtopiaOnX11

Approach to Qtopia on X11

Sending stuff upstream

- Once you have a set of patches and are confident these are good
- Search upstream, where to send them, how to send them.
- Send the patches upstream and wait
- The answer will be encouraging most of the time, take their input into account, redo your patches.
- git helps you there

Done. Questions?