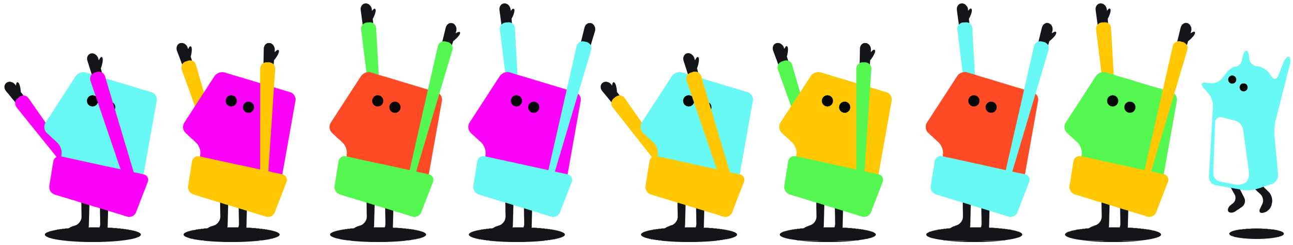
A developer's journey

MeeGo



The perfect toolbox

- OSC (Build Service client)
  - Version 0.130.1
- MIC2 (image creator)
  - Version 0.24.5
- Spectacle
  - Version 0.22
- MeeGo packaging tools
  - Version 0.6
- Flasher (flashing utility)
  - Version 3.5
- Build (script to build RPMs)
  - Version 2010.12.15

The dashboard

MeeGo Bugzilla - http://bugs.meego.com

MeeGo Build Service - http://build.meego.com/home/list_my

MeeGo community build service - https://build.pub.meego.com/

MeeGo Gitorious and the activity monitoring - http://meego.gitorious.org

MeeGo-commits mailing list - http://lists.meego.com/pipermail/meego-commits

OBS basics

OBS is an OS distribution development system. Regarding MeeGo N900-DE, it is an automated build system for building packages

OBS contains SW in the form of packages

- e.g. bash

Packages are kept under projects – similar to a directory structure

- e.g., Trunk/bash, Trunk:Testing/bash, home:mlehtonen/bash

OBS publishes built packages in repositories (for download and install)

More info: http://wiki.meego.com/Build_Infrastructure

OBS workflow for N900-DE

Workflow is simple in Community OBS

home:user → Project:DE:Trunk:Testing → Project:DE:Trunk

Workflow in MeeGo OBS

home:user → devel:topic → Trunk:Testing → Trunk

More information (MeeGo workflow)

- http://wiki.meego.com/Release_Engineering/Process

OSC – Build Service client

For interfacing with OBS and building packages in MeeGo, used for daily work in MeeGo

→ osc branch (bco)

→ osc checkout (co)

→ osc vc

→ osc diff (di)

→ osc build

→ osc chroot

→ osc commit (ci)

→ osc submitrequest (sr)

Good source for info and tutorial:

http://wiki.meego.com/Build_Infrastructure/Packagers_Developers#How_to_use_the_

OSC – Build Service client - Tips

Setup OSC

```
 $ osc
```

Edit ~/.oscrc and configure API url/host:

```
[https://api.pub.meego.com]
user = user_name
passx = user_password
aliases = community
```

Tips:

```
Extra-pkgs = vim rpmdevtools
http_headers = Authorization: Basic user_pass
 $ echo -n user_name:user_password | base 64
```

Platform development

OBS/OSC is the platform SDK in MeeGo

MeeGo SDK is targeted only for Apps and App-developers

OBS local buildroot can be used as a build environment

You can also directly chroot in meego rootfs images on your workstation

RPM packaging

Basics are simple

- One “recipe file”, .spec
- Source tarball
- Optionally patches or other additional files to add to the RPM package

.spec can be thought as a script file – easy to understand

RPM packages are easy to build in MeeGo

- Use osc command line client to build locally or remotely

More information

- http://wiki.meego.com/Packaging/Guidelines
- https://fedoraproject.org/wiki/How_to_create_an_RPM_package

RPM packaging – OBS package dir

OBS package directory should contain

- pristine source tarball from upstream repository
- .spec/.yaml file – rpm recipe
- .changes file – packaging changelog
- optionally, patches (to be submitted upstream)
- possibly, some additional files to be added in the package

Package (and tarball) version should always match the upstream

More information

- http://wiki.meego.com/Packaging/Guidelines
- https://fedoraproject.org/wiki/How_to_create_an_RPM_package

RPM packaging - Spectacle

Simple tools for maintaining rpm packages

- specify
  - Tool to generate/update spec files
  - Based on YAML format
- spec2spectacle
  - Tool to convert spec to YAML → new spec file

Usage of specify is encouraged

- Very easy to create new rpm packaging from scratch
- Makes maintenance of spec files a lot easier
- Automation and sanity checking

More information

- http://wiki.meego.com/Spectacle
- http://wiki.meego.com/Packaging/Tutorial

MeeGo-on-N900 – Installation

Enable dual boot (uBoot, NEW recommended way):

- http://wiki.meego.com/ARM/N900/Install/Dual_Boot

Get a micro SD card

Get ready-made MeeGo rootfs image

- http://repo.meego.com/MeeGo/builds/

Install rootfs to micro SD card:

- http://wiki.meego.com/ARM/N900/Install/MMC

(installing into internal eMMC not officially supported)

MeeGo-on-N900 – Tips

Enable USB networking and NAT in host

- http://wiki.meego.com/ARM/N900/Tips_and_Tricks/N900_USB_Networking

Install extra packages to N900 with zypper

```
$ ssh root@192.168.2.15
on device # zypper install $package
```

Instructions and help:

- http://wiki.meego.com/ARM/N900
- #meego-arm at Freenode IRC

Debugging

Use USB networking

Run binaries on-device

On-device gdb

Use SDK

Note: it requires to install debuginfo packages

Development flow example

What: fix a bug in PulseAudio.

Make a bugfix branch of the package and download its sources

```
$ osc branch Trunk:Testing pulseaudio
$ osc checkout home:mlehtonen:branches:Trunk:Testing pulseaudio
```

Make the changes/fixes and build the package

```
$ cd home:mlehtonen:branches:Trunk:Testing/pulseaudio
<do fixes>
$ vim pulseaudio.changes
$ osc build –-no-verify standard armv7el    ← build locally on your workstation
$ osc ci                                    ← submit to OBS server for remote build
```

Development flow example

Test the package

```
$ scp /var/tmp/build-root-armv7el-standard/home/abuild/rpmbuild/RPMS/armv7l/pulseaudio-0.9.19-0.armv7l.rpm root@192.168.2.15:
$ ssh root@192.168.2.15
device# rpm -Uvh pulseaudio-0.9.19-0.armv7el.rpm
<run and debug>
```

After verifying your changes, submit changes back to Trunk:Testing

```
$ osc submitrequest
```

Update status and comment on bugzilla

Questions & Answers

Thanks!