
Emdebian

Embedded development the Debian way.



Overview

- What is Emdebian?
- Short history
- Technical challenges
- Existing approach
- How does it work?
- Future developments
- How to help

What is Emdebian?

- Official sub-project of Debian
- Is all about bringing Debian to embedded systems
- Tries to take advantage from the existing multi-architecture support
- providing Free / community supported embedded development tools

History

- Project started in 2000 by Frank Smith
- Produced Embedsys and a toolchain
- In 2001 update to CML2 and new project lead
- In 2003 Embedsys updated to Kconfig
- In 2004 new start, new approach with the Stag Framework
- Now: growing interest and active development

Short intro to Debian

- The main non-commercial GNU/Linux distribution
- Known for its package management
- Supports several architectures
- Huge number of packages (8740+)
- Strict quality control

Technical challenges

- Debian is mostly about native development
- The size of the binaries is not taken into account
- Lots of documentation that is not needed on the embedded system is included
- Efficient cross-compilation scheme?
- Dealing with optimisations
- Software management on the target/host ?
- Using other C libraries than glibc

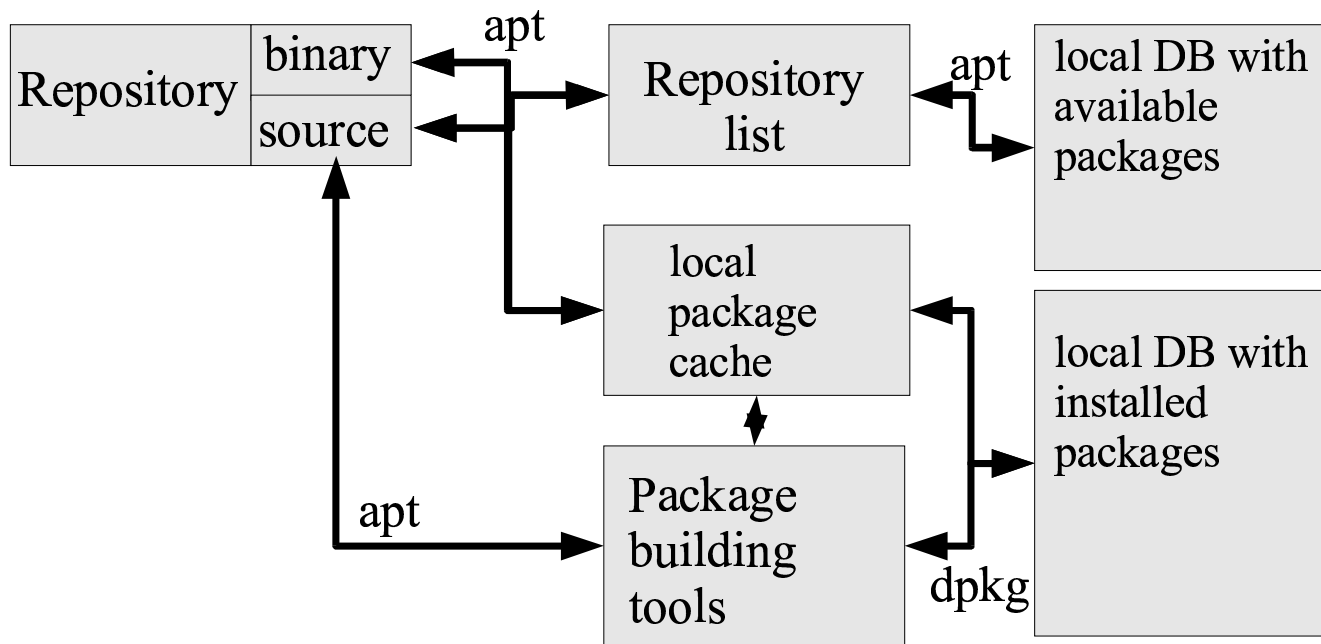
Current approach

- Trying to re-use the package management system
 - Adding support for flexible cross-compilation
 - Taking size into account and removing documentation
 - Binary packages should be installable on the development host and the target
- => Redefined source package

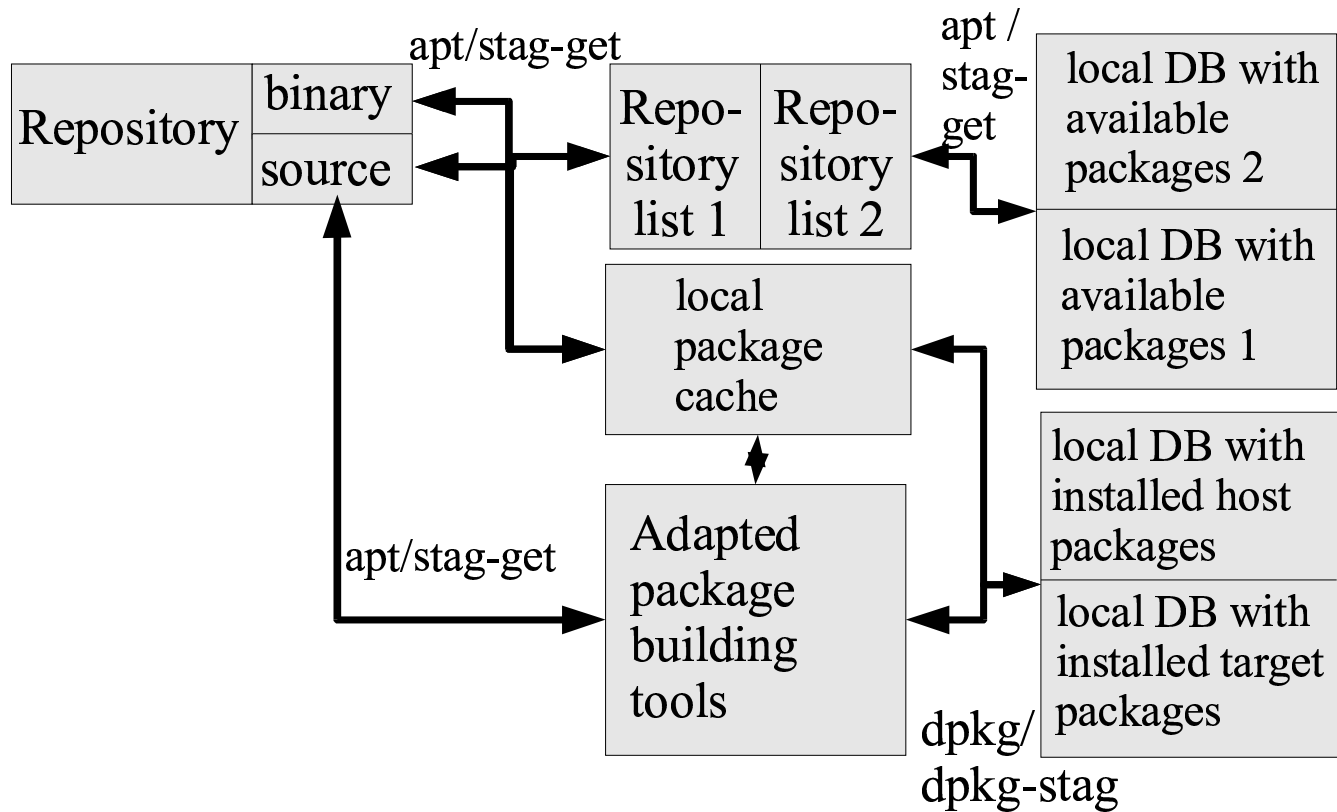
How?

- Main component adapted: dpkg-cross
- Dpkg-cross fits into the package management system
- “Forcing” the use of a cross-compiler toolchain instead of native gcc
- Extending the Debian source packages with info for cross-compilation
- Adapted tools to cope with that information

Normal package management system



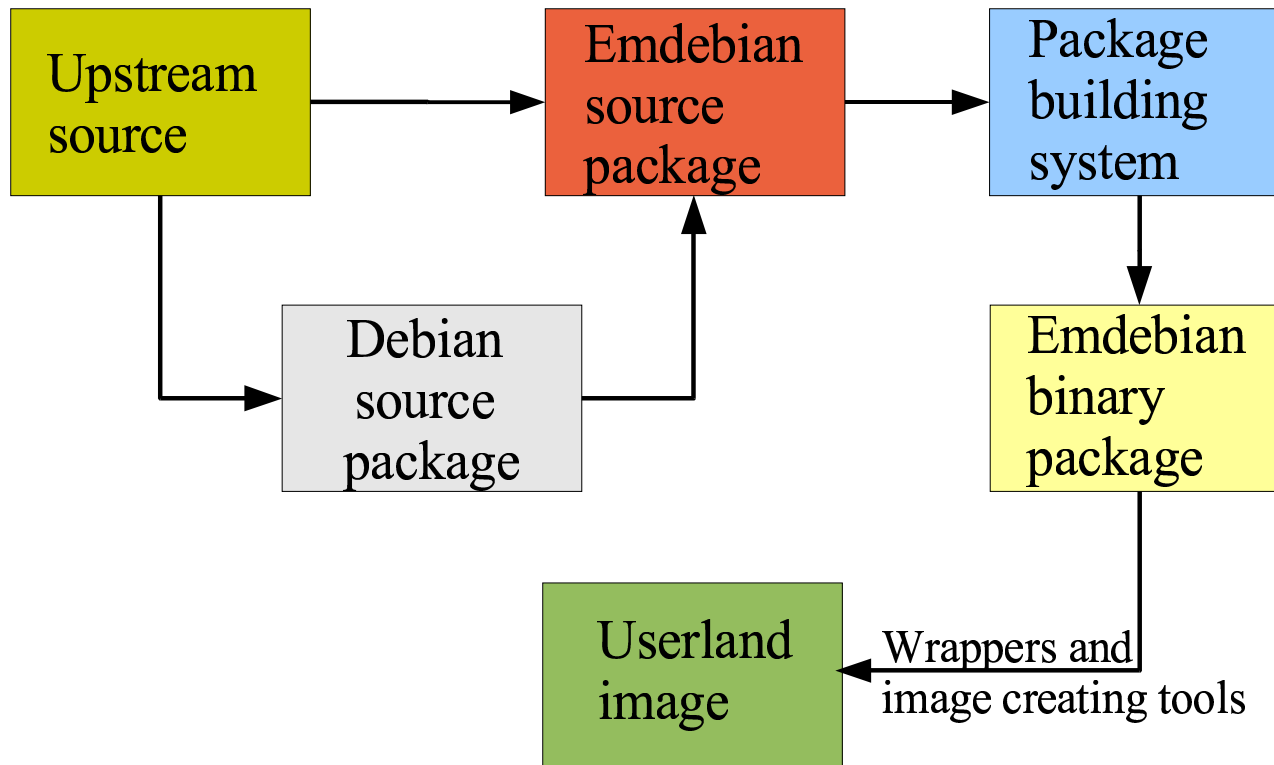
Adapted scheme



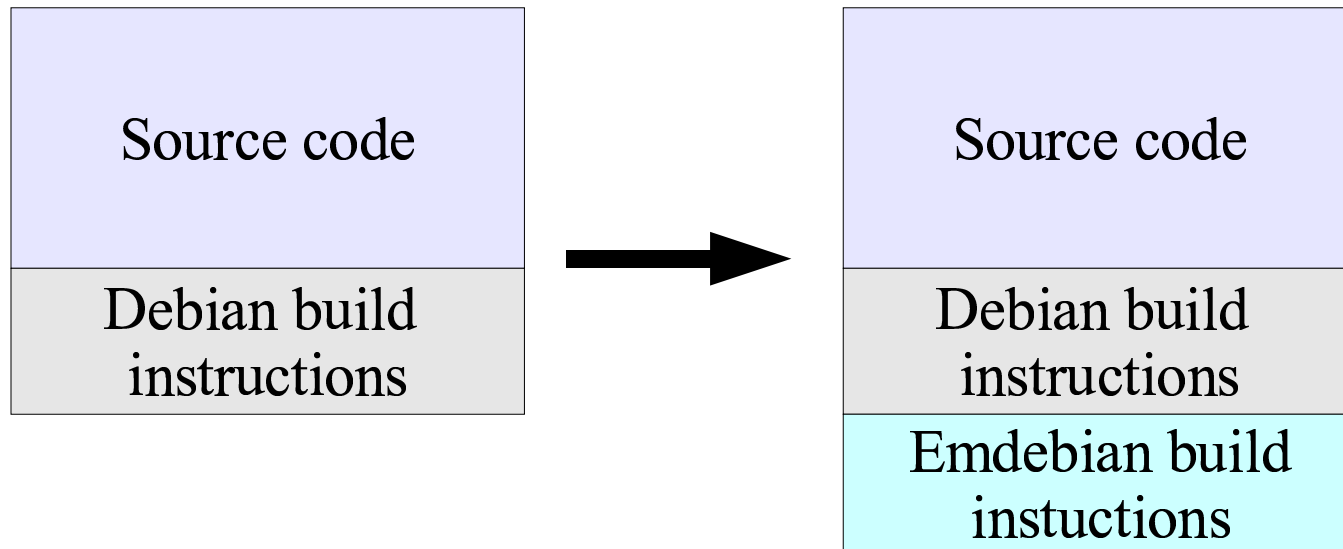
How it works

- All parameters are controlled in one file (/etc/dpkg/cross-compile)
- Packages are build in just the same way as normal Debian ones
- Wrappers for installing and managing the packages on the development host
- Managed by ipkg or full dpkg on the target

Emdebian work-flow



Revised package format



Basic things to know about the Debian source package

- Contains the source code (of course :-)
- Has a debian subdirectory
- Inhere we find several files, the most important are:
 - rules: used to do the actual building of the software
 - control: contains the description and dependency information
 - Changelog: Well this one is obvious
 - postinst, postem, preinst, prerm,... : Little scripts to perform post installation or removal operations

Detailed package changes

- The package stays essentially the same
- Emdebian subdir added with specific build information
- The new build information is similar to the Debian build info.
- Differences are: no docs, -Os optimization, package naming,...
- Use of adapted debhelper tools in rules file

Role of /etc/dpkg/cross-compile

- Contains the necessary parameters
- `crossbase` and `crossdir` to specify the working directories of the cross-compiler
- `CROSSPREFIX` is used to specify the right cross-compiler and the correct strip binary
- `LIBC` is used for the naming scheme, to keep apart packages with the same architecture but a different C-library
- `CONFIG` specifies the use of a default or manual configuration
- `emdebian` makes the package tools use the emdebian build information

Using Emdebian

A number of “easy” steps:

1. Install a cross-compiling toolchain
2. Edit the `/etc/dpkg/cross-compile` file for your use and toolchain (the most important parameters are LIBC, CONFIG and CROSSPREFIX)
3. Fetch a Emdebian source package(s)
4. Extract it (them) in your development location
5. Change into the new dir and execute “`dpkg-buildpackage -b -a<arch> -D -Emdb`”
6. Install the new binary package(s) with “`dpkg-stag`”
7. Make an image of your new root-filesystem and upload it to your board

Current and future development

- Dpkg-cross is being reviewed and updated
- New Debianized toolchains
- Package list and adapted packages
- Cross-compiling with distcc
- Adding Qemu to solve binary tests when cross-compiling
- Better wrappers
- IDE user interface

How to help / where to get help?

- Subscribe to the mailing list (debian-embedded@lists.debian.org)
- IRC channel on freenode (#emdebian)
- www.emdebian.org & stag.mind.be
- Convince package maintainers to add Emdebian build information
- Test it on different platforms and submit problem descriptions (if any)

Time for questions

Just ask away :-)

A demonstration or further discussion is also possible after the presentation, just feel free to ask me.

Thanks



mind
embedded development
www.mind.be



www.aleph1.co.uk

Many individuals : Peter Naulls, Jon Masters, Justin Cormack,
Nikita Y.Youshchenko, Raphael Bossek, Allen Curtis,
Benjamin Henrion and anyone I might have forgotten.