

# Creating Qt Applications for Maemo, Beta

Course Overview

Forum **Nokia**

**NOKIA**


Forum Nokia

# License


Copyright © 2010 Nokia Corporation and/or its subsidiary(-ies).

Nokia, the Nokia logo, Qt, and the Qt logo are trademarks of Nokia Corporation and/or its subsidiary(-ies) in Finland and other countries. Additional company and product names are the property of their respective owners and may be trademarks or registered trademarks of the individual companies and are respectfully acknowledged. For its Qt products, Nokia operates under a policy of continuous development. Therefore, we reserve the right to make changes and improvements to any of the products described herein without prior notice. No warranty, express or implied is made about the accuracy and/or quality of the information contained herein. Under no circumstances shall Nokia Corporation or any of its subsidiary(-ies) be responsible for any loss of date, or income, or any direct, special, incidental, consequential or indirect damages whatsoever.

This document is licensed under the Creative Commons Attribution-Noncommercial-Share Alike 2.5 License.



For more information, see <http://creativecommons.org/licenses/by-nc-sa/2.5/legalcode> for the full terms of the license.



## Course Objectives and Prerequisites

Forum Nokia

- Objective is to study developing Qt applications for Maemo 5 platform
  - Get familiar with Maemo platform and the development tools
  - Learn porting Qt applications from other platforms to Maemo
  - Learn fine-tuning the UI for Maemo
  - Learn Qt features only available in Maemo
- It is assumed that the course attendant has
  - Basic knowledge of Qt
  - Some previous Linux or Maemo knowledge is a big advantage

NOKIA

<h2>Table of Contents</h2>		Forum Nokia
• Lecture 1:		
• Maemo Introduction		
• Development Environment		
• Running Qt Apps in Maemo Devices		
• Lecture 2:		
• Widgets and Appearance		
• Additional Qt for Maemo Topics		
• Lecture 3:		
• Qt Mobility API		
• Current Status and Summary		
• Exercises		
• Available as a separate exercises document		

NOKIA

# Creating Qt Applications for Maemo

Maemo Introduction

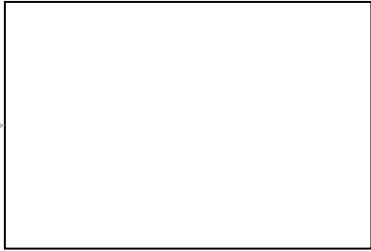

Forum **Nokia**


**NOKIA**

# What is Maemo?

Powered by Nokia:  
<http://maemo.nokia.com/>

Powered by the community:  
<http://maemo.org/>





Maemo is an open source development platform for Linux-based mobile computers (previously known as Internet Tablets). It is built from widely used open source components with additional tweaks to improve the use of handheld devices. Maemo is driven by Nokia as well as an active open source developer community.

For a programmer, these devices open a lot of possibilities because they are largely based on free software, so you can use tools that you use in normal software development for free and open systems. The tools, libraries, and development processes that are used in Maemo are equally used and applied in the desktop application arena as well as for building server software. This is thanks to the GNU project which has implemented much of the tools infrastructure in a highly portable way.

As opposed to desktop Linux, Maemo is considered a single-user system. The security model in Maemo is focused on protecting the user from remote attacks rather than other users. Another major difference compared to desktop Linux is that Maemo uses a lightweight executable called BusyBox as a replacement for the GNU shell and essential command line tools.

Traditionally development for Maemo has been done with C using the GTK+ widget toolkit. Thanks to the developer community several other languages can be used as well, including C++, Python, Ruby, Mono and even Java™. Qt has been previously available as a community port, and for Maemo 5 and official Qt 4.6 port has been made.

## Maemo Devices and OS Versions

Forum Nokia

- **Nokia 770 Internet Tablet**
  - Based on Internet Tablet OS 2006 edition and Maemo platform v2.2 – Gregale
- **Nokia N800**
  - Based on Internet Tablet OS 2007/2008 edition and Maemo platform v3.2/v4.0 – Bora
  - GTK+ based UI framework
- **Nokia N810**
  - Based on Internet Tablet OS 2008 edition and Maemo platform v4.0/v4.1 – Chinook, Diablo
  - Any UI framework running on the top of X11 (Qt libraries, Java)
- **Nokia N900**
  - **Based on OS 2009 edition and Maemo platform version 5 – Fremantle**
  - **Phone functions (mobile computer)**
  - **Official Qt 4.6 port to Maemo 5 announced by Nokia**

NOKIA

This training material is based on the latest Maemo platform, which is version 5.0 released along with the Nokia N900 device. As a change to the previous versions, Maemo 5 contains also phone functions. A major change from Qt developers' point of view is that Nokia has officially announced to provide official Qt 4.6 port to Maemo5. Code examples in this course use Qt 4.6 unless otherwise stated.

An excerpt from the press release:

*"NOKIA PRESS RELEASE October 9, 2009*

*Nokia announces official Qt port to Maemo 5*

*Developers will be able to target the Nokia N900 and upcoming Maemo and Symbian devices.*

*Espoo, Finland - Nokia today announced the creation of the official Qt port to Maemo 5 and published an initial Technology Preview release. This port to Maemo 5 means that developers can now use Qt to target the upcoming Nokia N900 device and also ensures that applications can be easily ported to all Qt's supported platforms including the next Maemo 6 release as well as Symbian and Windows Mobile.*

*The Qt port to Maemo 5 is designed specifically to work within the Maemo 5 environment, which will power the upcoming Nokia N900 device. The port itself will be based on Qt's upcoming version 4.6 and is scheduled for final release in Q1 of 2010...."*

# N900 Hardware

Forum Nokia

N900 Features	
Processor	OMAP 3430 with 600 MHz clock frequency.
Memory	Up to 1GB of application memory
Storage	32 GB, extendable up to 48 GB
Display	800x480 (3.5") touchscreen
Camera	5MPx with autofocus, WVGA video
Graphics	Hardware-accelerated with OpenGL ES 2.0 support
Connectivity	Wi-Fi, HSPA, quad band GSM, USB, Bluetooth
Other	A-GPS, accelerometer

NOKIA

**Announced Date**

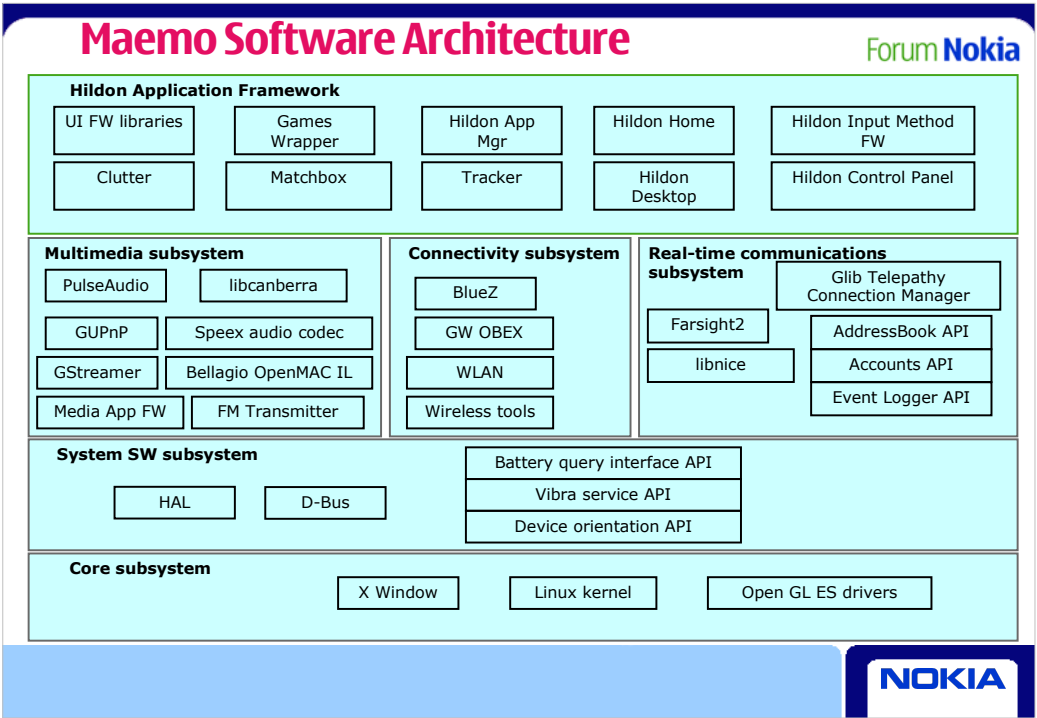
27 August 2009

**Device Description:**

*The Nokia N900 brings the power of computer to mobile devices. It's a high performance mobile computer with advanced Maemo 5 software on Linux. Explore Nokia N900 features like multitasking on a live Dashboard, wide screen video capture, 3,5" WVGA touch display (800 x 480 pixels), QWERTY keyboard and personalisable panorama desktop. The Nokia N900 also features 1 GB of application memory and up to 48GB of storage. Maemo browser is powered by Mozilla Technology and includes Adobe Flash 9.4 and full AJAX support. Wi-Fi and 3.5G connectivity provides fast access to Internet. Take high quality photos and wide screen videos using 5Mpix camera with Carl Zeiss optics. Tag photos and browse them using the tag cloud, and see where images were taken with the automatic geotagging. Merge your phone book, Skype contacts and other contacts into an all-in-one address book. And naturally, N900 has a phone application for voice calls.*

For more information, see:  
<http://www.forum.nokia.com/devices/N900/>





With Qt APIs, the underlying platform architecture should not be of great concern. However, when implementing advanced functionality that Qt does not cover, it may become necessary to have understanding about the system and its services.

Notice that this architecture diagram is not complete but just an exemplification. Refer to the Maemo Developer Guide for more detailed information.

**Application framework:**

The purpose of an application framework is to help application development by providing a standard structure for an application. Applications that have a graphical user interface tend to have a similar structure, for example the event-driven run time model. The application framework of Maemo is called Hildon. Implementing Qt applications coherent with other Maemo applications involves conforming to Hildon UI style. This will be studied later during the course.

**Middleware libraries and frameworks:**

Maemo contains a variety of open source libraries and frameworks. They are commonly used building blocks of different Linux distributions. Maemo is built on the GNOME mobile platform and the set of libraries included in it. Typically, Maemo developers also utilize the services provided by these libraries in their own applications. As for any other Linux distributions, it is naturally possible to manage the components and install new ones to the device.

**Core subsystem & System SW:**

Linux 2.6 kernel handles hardware events, memory allocation, process creation, and everything that you expect from a modern multi-tasking UNIX-like kernel. D-Bus, is a daemon that enables related processes pass events to each other. A daemon is a process that runs in the background, similar to a service in Microsoft Windows. The D-Bus daemon also passes important events such as “Battery low” from the core system to applications. HAL provides a hardware abstraction layer which allows applications to discover and use the hardware through a simple, portable and abstract API, regardless of the type of the underlying hardware. The X Window System is an architecture-independent client/server system that allows multiple programs to interact with the user via a graphical screen, keyboard, and other input devices such as a touchscreen. We will take a closer look at the X Window System later in this course.

# Creating Qt Applications for Maemo

Development Environment


Forum **Nokia**

**NOKIA**

## Development Tools Overview

Forum Nokia

- Maemo development environment consists of:
  - Desktop machine with Linux, Mac OS X or Windows operating system
    - Linux (32 bit and 64 bit)
    - Windows (32 bit and 64 bit)
    - Mac OS-X (10.5 and following)
  - Nokia Qt SDK, consisting of the following tools for Maemo devices:
    - Qt Creator
    - Qt Simulator
    - MADDE
  - Needed installation packages for the N900
    - rootsh – For obtaining root access
    - mad-developer – Provides robust way to ease developer access to Maemo device
  - Maemo PC Connectivity for rapid deployment (optional)



The Nokia Qt SDK requires the firmware version of the device to be PR 1.2

- All-in-one package for Qt development for Windows, Maemo and Symbian platforms
- Qt Simulator is scriptable simulator for testing purposes
  - Events and various situations can be simulated, such as low battery level, location change, etc.
- Qt Creator is an IDE for developing Qt programs
  - An integrated support for building and running programs in desktop environment, Qt Emulator as well as in Maemo and Symbian devices

## Nokia Qt SDK – MADDE

Forum **Nokia**

- “Maemo Application Development and Debugging Environment”
- MADDE provides command-line cross-compiling
- Has multi-platform support (Linux, Windows, Mac OS X)
- Configurable for different targets & toolchains
- Client for the device to simplify the development process (mad-developer)
- For more information, see [MADDE wiki page](#)

**NOKIA**

# Nokia Qt SDK – Qt Simulator


Forum Nokia

- Simulator for testing Qt programs for mobile devices
- Supports also Qt Mobility APIs
- User can simulate different battery levels, charging states, profiles, storage, network connections, locations, contacts, messaging and sensors
- Simulated events can be launched manually or by using a script
- Different skin types include Maemo, Symbian Touch and Symbian NonTouch




NOKIA

## Alternative Development Tools



- Previously, the Maemo development was done with the following tools:
  - Desktop machine with Linux, Mac OS X or Windows operating system
    - True Debian-based Linux PC as a recommended native environment
    - Other Linuxes, Windows and Mac OS X can be used with virtual Linux images
    - Maemo SDK Virtual Image recommended for fast kick-start
  - Scratchbox
    - Cross-compilation toolkit
  - Maemo SDK
    - Rootstraps for x86 and ARMEL targets inside Scratchbox
  - Xephyr
    - Pseudo X server for device screen emulation
  - IDE and other optional tools
- Detailed installation instructions are available in the [Maemo wiki](#)



The development environment for Maemo running on the desktop is called Maemo SDK, which only installs and runs on a Linux operating system. Supported Linux distributions are Debian and Ubuntu, but installing Maemo SDK is also possible for other distributions. On other operating systems such as Windows, virtualization software can be used to provide a working Linux environment in a very easy manner. Maemo SDK Virtual Image is a tailor-made solution for Maemo developers, it includes all necessary tools preinstalled and configured allowing an immediate start of development.

Maemo SDK establishes a sandboxed development environment largely built on a tool called Scratchbox. It allows compiling the software for the host's x86 architecture and for the ARMEL architecture of the actual device. A certain build configuration is called a "target". Maemo SDK contains so-called "rootstraps" for both x86 and ARMEL targets inside Scratchbox. A rootstrap represents the root file system of a true Maemo device thereby appearing as a sandbox environment to the developer.

Xephyr X server is a separate tool which provides a graphical user interface where the device screen can be emulated on PC.

Maemo SDK does not contain an integrated development environment. ESbox is a dedicated Eclipse-based IDE Maemo development. Qt developers may decide to use Qt Creator also for Maemo development.

Maemo 5 SDK installation guidelines:

[http://wiki.maemo.org/Documentation/Maemo\\_5\\_Final\\_SDK\\_Installation](http://wiki.maemo.org/Documentation/Maemo_5_Final_SDK_Installation)

## Alternative Development Tools – Scratchbox

Forum Nokia

- Cross-compilation in Linux is difficult – well is it actually?
  - Partly because of autotools – which should be helpful
- Some Linux distributors solve the problem by not supporting cross-compilation
- Scratchbox solves the problem by totally isolating host and target environments
  - Supports two separate targets: X86 and ARMEL
  - <http://www.scratchbox.org>
- Provides also emulation of Maemo
- Currently Maemo officially supports Scratchbox 1. Development around another version, Scratchbox 2, is ongoing.

NOKIA

Scratchbox 1 (SB1) requires a copy of each file in the isolated file system

- Rootstrap – like a “standard” Linux file system
- Safe for dummies
- Impossible to overwrite system files – user may only break the SB environment
- Large disk image
- SB supports only vi and pico
  - Install another copy of your favorite editor or edit outside SB

Scratchbox 2 (SB2) shares as many system files as possible

- Not for dummies
- More efficient disk usage
- Easier integration of tools – is it difficult in SB1?

Autotools-based build scripts can be run on Scratchbox without modification when building for the target

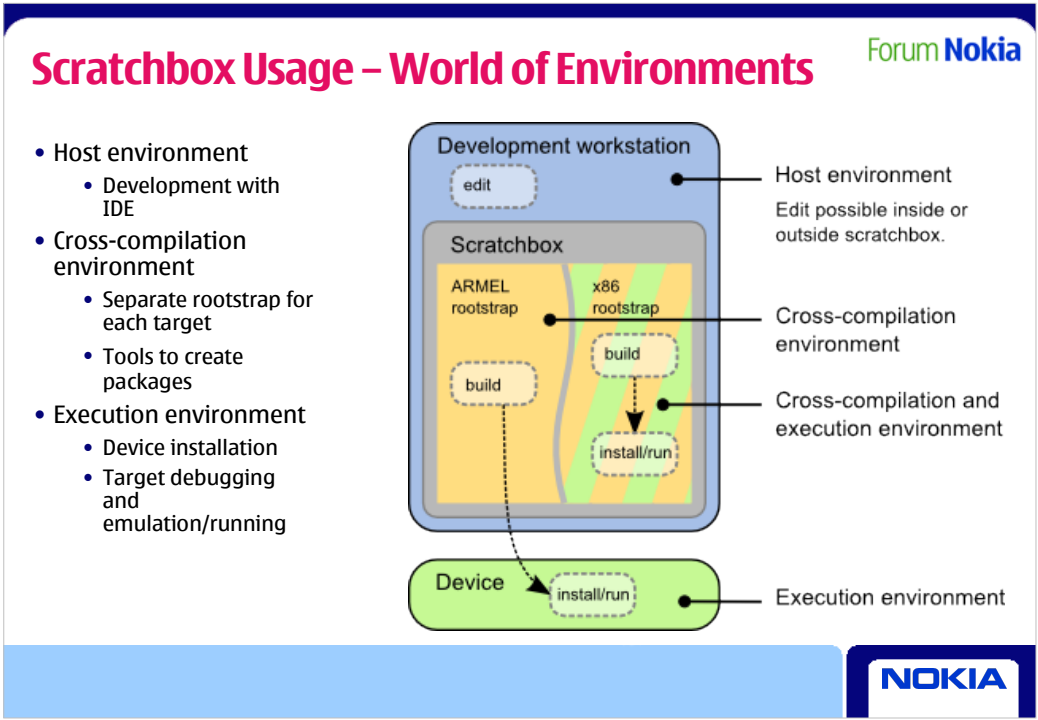


## Scratchbox Usage – Basics

Forum **Nokia**

- Start it by typing the command
  - `scratchbox`
- Configure targets
  - `[sbox-FREMANTLE_ARMEL: ~]: sb-menu`
- Switch between targets with `sb-menu` or by typing:
  - `[sbox-FREMANTLE_ARMEL: ~]: sb-conf select FREMANTLE_X86`
- Otherwise work in a similar way as outside SB
- Single source concept
  - Single source – multiple targets

**NOKIA**



In SB1, you work inside Scratchbox  
In SB2, you give sb2 command in front of each SB command

## Scratchbox Usage – Installing Qt Libraries

Forum **Nokia**

- Scratchbox used
- You may use ESbox, but you need to write build script yourself
- In Maemo SDK Virtual Image, the scratchbox does not initialize resolv.conf properly, as said in [here](#). Use the following script to set the /etc/resolv.conf of scratchbox to obtain access to repositories:
  - `maemo@maemo-desktop:~$ /scratchbox/sbin/sboc_ctl restart`
- Scratchbox contains Qt 4.6 packages pre-installed. To manually install them, use the following commands. Remember to install them for both x86 and armel targets.
  - `fakeroot apt-get update`
  - `fakeroot apt-get install libqt4-dev`
- Installing libqt4-dev manually installs also the Qt Mobility API packages

**NOKIA**

Log in to Scratchbox. Bear in mind that x86 and ARMEL are separate rootstraps, which means that installation needs to be done for both. For Nokia N900 PR1.2, the Qt is preinstalled. Qt 4.6 for Maemo can also be downloaded via the Qt website: <http://qt.nokia.com/downloads>. If you have any network problem during the installation, please check the following issues:

**DNS problems:** From a normal Linux shell check if /scratchbox/etc/resolv.conf is properly configured (it is supposed that your scratchbox is installed inside /scratchbox/). If it is wrong, you can copy your Linux resolve.conf using the following command: `cat /etc/resolv.conf | sudo tee /scratchbox/etc/resolv.conf`

**Proxy problems:** If you are getting an error about resolving “nokes.nokia.com”, you could try removing that proxy in a file named “99proxy” under /etc/apt/apt.conf.d of each of your Scratchbox targets.

On the other hand, if you are using some proxy, it is necessary to configure it for proper APT operation. Add the following information in your “99proxy” file mentioned above, in each of your Scratchbox targets:

Acquire::http::Proxy "http://proxy\_user:user\_passwd@proxy\_address:proxy\_port";

Just replace proxy\_user, user\_passwd, proxy\_address and proxy\_port by your own proxy settings.

For newer Qt versions you need to have “Extras” and “Extras-Devel” configured in each of your Scratchbox target's /etc/apt/sources.list:

`deb http://repository.maemo.org/extras/ fremantle free non-free`

`deb-src http://repository.maemo.org/extras/ fremantle free`

`deb http://repository.maemo.org/extras-devel/ fremantle free non-free`

`deb-src http://repository.maemo.org/extras-devel/ fremantle free`

Keep in mind that “Extras-Devel” contains software that can still be unstable and is only recommended for developers and adventurous users. Remember to update the package manager cache after making any modifications to the repository list.

# Creating Qt Applications for Maemo

Running Qt Apps in Maemo Devices

Forum **Nokia**

**NOKIA**

## Building Qt Applications

- For a created Qt project, you have to set up the needed targets
  - Targets can be added from the Qt Creator's Projects view, in the Targets section of the corresponding project tab, by clicking the plus sign and choosing the Maemo
  - Remember to choose the Qt version which has been installed with the Nokia Qt SDK in the target's Build setup view
  - The Qt Creator settings are stored in a <project name>.pro.user file
- Set the Maemo target as default from the list of targets

main.cpp - appchooser - Qt Creator

File Edit Build Debug Tools Window Help

MyMobileQtApp appchooser

TARGETS Editor Settings Dependencies

Desktop Build Run

Build Settings

Edit Build Configuration: Debug Add

General

Help

Project MyMobileQtApp

Desktop

Build: Debug Run: MyMobileQtApp

Maemo

Build: Debug Run: MyMobileQtApp on Maemo di

NOKIA

Shadow build is used to build binaries from the sources in a different folder than where the original files are located. This makes it possible to compile the program for many different platforms from single source files without copying them.

Copyright © 2010 Nokia Corporation.

21

## Developing with Device (N900)

Forum Nokia

- Setting up basic development environment
  - Add Extras and Extras-devel to Application manager catalogues
    - See notes for repository details
  - Install mad-developer from Extras repository for quick USB access to the device
  - Install rootsh from Extras repository to obtain root privileges (see [details](#))

New catalogue

Catalogue name: extras devel

Web address: y.maemo.org/extras-devel

Distribution: fremantle

Components: free non-free

☐ Disabled

Save

NOKIA

Installing rootsh from Extras repository is the easiest way to obtain root privileges. The repository details for Extras[-devel] are:

- Web address: [http://repository.maemo.org/extras\[-devel\]](http://repository.maemo.org/extras[-devel])
- Distribution: fremantle
- Components: free non-free

Use the Application Manager to add repositories and install packages. In the Application Manager, click on the “Application catalogues” item in the application’s menu. Click on the “Extras” “Extras-Devel” catalogue entry to view their details and make sure that they are enabled. If they are non-existing, you can create them by hand.

When needing to use command line tools with the device, open the X Terminal application, and type “sudo gainroot” to enable root shell. Then you can run for instance apt-get, dpkg and other tools as usually.

Note that mad-developer needs quite large amount of data to be downloaded. If no WLAN network is available, you could download mad-developer and the needed openssh-server, libfuse2, fuse-utils, libqt4-core, libqt4-gui, usb-network-modules and gdb debian (.deb) packages, which are available in internet. For example, see <http://repository.maemo.org/extras/pool/fremantle/free/m/mad-developer/>.

See <http://maemo.org/packages/> for a package search functionality.

## Developing with Device (N900)

Forum **Nokia**


- Useful information about the device:
  - Folder 'Nokia N900' equals to '/home/user/MyDocs'.
  - Files stored in 'MyDocs' will not be available in “Mass Storage mode” when connected to the computer via USB (from Maemo Wiki [FAQ](#)). To connect with the “PC Suite Mode”, you need the Nokia PC Suite or OVI Suite installed.
  - The images taken with native camera application go into ~/MyDocs/DCIM folder (which is equal to “Camera” folder), not into ~/MyDocs/.images folder!
  - Ctrl+Shift+X opens the terminal, Ctrl+Fn+P takes a screenshot and places it into a folder MyDocs/.images/Screenshots. For more shortcuts, see [N900 shortcuts and gestures](#)

**NOKIA**

## Installing Qt Libraries on a Device (N900)

Forum Nokia

- N900 PR 1.2 has the Qt 4.6.2 packages pre-installed.
- Alternatively, binaries can be installed with the terminal application. For example:
  - `sudo gainroot`
  - `apt-get update`
  - `apt-get install libqt4-core`
- Qt Mobility API binaries are installed automatically when installing packages depending on them. One such package is the qt-mobility-examples package:
  - `sudo gainroot`
  - `apt-get update`
  - `apt-get install qt-mobility-examples`



You need to make sure that the Qt packages are installed also on the actual device.

There can also be other Qt binaries available. To install them you might need to uninstall previous version of Qt with `apt-get remove libqt4-*` or similar command.

You deduce what Qt Mobility API packages are available by searching for the qt-mobility-examples dependencies with either

`apt-cache show qt-mobility-examples`

or by searching it from the <http://maemo.org/packages/>.



## Setting Up the USB Networking

Forum Nokia

- On the device, in the Mad Developer:
  - In Windows: click on "Manage USB", select "Load g\_ether", and then click "Close".
  - Following click "Edit" in the usb0 row and confirm with "Configure".
  - usb0 IP address should now be 192.168.2.15
  - Now select "Developer Password" to generate a password for a freshly created user called "developer"
- Connect the device with USB, and choose "PC suite mode" for the mode
- Setting up USB networking
  - In Windows, change the IP to be set statically for the USB network and enter the following values:
    - IP Address: 192.168.14
    - SubnetMask: 255.255.255.0
  - In Linux, run the command `ifconfig usb0 192.168.2.14 up`
  - For Mac OS X, see [http://wiki.maemo.org/MADDE/QtCreator\\_integration\\_for\\_Mac](http://wiki.maemo.org/MADDE/QtCreator_integration_for_Mac)

NOKIA

To use network connection and connect the device at the same time in Windows, you must install Maemo PC Connectivity. This is the preferred way. See

[http://wiki.maemo.org/MADDE/QtCreator\\_integration\\_for\\_windows](http://wiki.maemo.org/MADDE/QtCreator_integration_for_windows)

[http://wiki.maemo.org/MADDE/QtCreator\\_integration\\_for\\_linux](http://wiki.maemo.org/MADDE/QtCreator_integration_for_linux)

[http://wiki.maemo.org/MADDE/QtCreator\\_integration\\_for\\_Mac](http://wiki.maemo.org/MADDE/QtCreator_integration_for_Mac)

## Running the Program with Qt Creator

Forum Nokia

- To run program, you need to setup the device configuration in project's run section in the projects view
- Click Manage device configurations in the Run setup view or go to Tools > Options... > Qt4 > Maemo devices to open Maemo device configuration view
  - Add a Maemo device configuration and set the password as given by the Mad Developer
  - Test the connection by clicking the Test button
- Now you can run the program by clicking the Run button or with Ctrl+R keyboard shortcut

Desktop

Maemo

Build Run Build Run

Run Settings

Run configuration: MyMobileQtApp on Maemo device Add Remove

Run configuration name: MyMobileQtApp on Maemo device

Device Configuration: Maemo device 1 [Manage device configurations](#)

Executable: C:\Work\Kinse\examples\MyMobileQtApp\MyMobileQtApp

Arguments:

Debugger: c:\miscprograms\madde\targets\freemantle-qt-0951\bin\gdb [Set Maemo Debugger](#)

NOKIA

## Scratchbox Usage – Building

Forum Nokia

- You can build, create package files and emulate the Maemo environment with the help of Scratchbox and Xephyr X Server
- Building in Scratchbox
  - `qmake -project`
    - Creates the `.pro` project file (if not already created)
  - `qmake`
    - Creates Makefile with Qt add-ons (`.moc` files, `.rsc` files etc.)
  - `make`
    - Builds the project

NOKIA

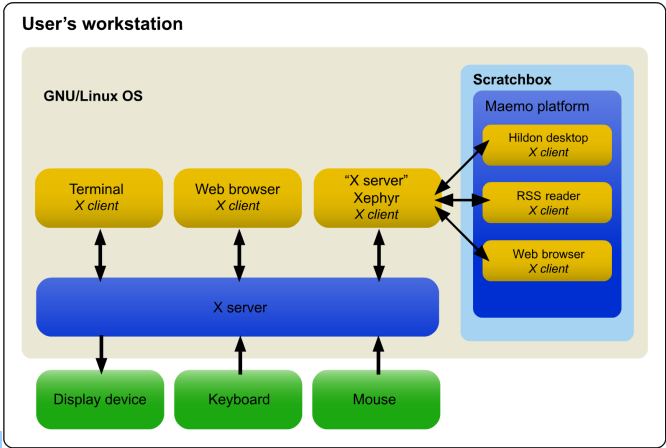
Experimental Qt versions are normally among the optional files. If `qmake` is not found or linking fails, because libraries are not found, add the following environment variables

```
$ export PATH=$PATH:<QT install folder>/bin
```

```
$ export LD_LIBRARY_PATH=$LD_LIBRARY_PATH:<QT install folder>/lib
```

## Scratchbox Usage – X Server

- X server allows applications to create windows to which they can draw on the screen
- Applications are X server clients
- X server also provides key press and pointer events to applications
- Maemo applications require a pseudo X server to be installed
  - It acts as an X client to a real X server
- Xephyr X server is typically used with Maemo SDK
  - Kdrive-based X server capable of emulating 16-bit color depth for clients



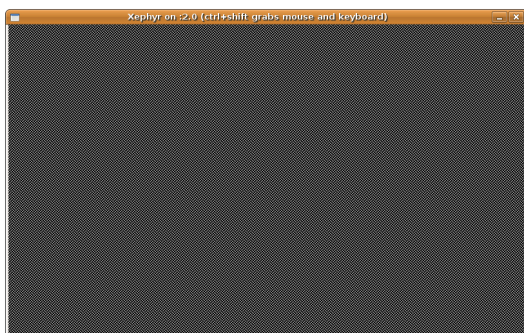
NOKIA

Maemo applications require a pseudo X server to be installed in your Linux system. The pseudo X server is the client of the real X server and it emulates the device screen in a PC. The X server allows applications to draw on the screen and provides events, such as key presses, back to applications. The Xserver used in this presentation is Xephyr.

## Scratchbox Usage – X Server (2)

Forum Nokia

- If Xephyr is not yet installed, you should install it (outside Scratchbox):
  - `$ apt-get install xserver-xephyr`
- Outside Scratchbox, say
  - `$ Xephyr :2 -host-cursor -screen 800x480x16 -dpi 96 -ac &`
- In Scratchbox, direct the display to window 2 by typing
  - `[sbox-FREMANTLE_X86: ~]: export DISPLAY=:2`



NOKIA

Install Xephyr using the command “`sudo apt-get install xserver-xephyr`”. After that, start the X server using the commands shown on the slide. These commands are given to your host Linux system, so do not give them inside Scratchbox. The screen option defines the size of the X server window. The number 16 means the color depth. Option `-ac` allows any client to access the server, and `:2` means the window number. Now the X server is up and running, but you still need to redirect Scratchbox to the X server. So, set the `DISPLAY` environment variable to redirect Scratchbox to Xephyr. This command is given inside Scratchbox, as can be seen from the changed prompt on the slide.

## Scratchbox Usage – Application Emulation


Forum Nokia

- Launch the application framework
  - [sbox-FREMANTLE\_X86: ~]: `af-sb-init.sh start`
  - Hint! If you cannot launch application framework, the process may be running already. Give a command `af-sb-init.sh stop` in that case
- Run standalone shell (themes, fonts will work)
  - [sbox-FREMANTLE\_X86: ~]: `run-standalone.sh ./myqtapplication`




NOKIA

## Debian Packages



- Maemo applications are distributed as Debian packages
  - So Qt apps in Maemo as well
- Packaging a Qt application into Debian package is relatively straightforward
- Debian package file structure for Qt programs is as follows:
  - `<package>-<version>/` (“myqtapplication-1.0/” for myqtapplication, version 1.0)
    - `<package>.pro` – rootstrap file to launch the actual .pro file in src folder
    - `debian/` – files needed for packaging a debian package.
      - Contains at least files named control (specifies package data, dependencies, binaries), changelog, compat and rules (makefile containing commands to build the package).
    - `src/` – sources
  - other arbitrary files and folders, like `data/`, which usually contains icons, .desktop and other files for packages



The main packaging toolset in Debian is called “APT”, and the most interesting utility within APT is the “apt-get” command. It reads package dependency and conflict information and tries to solve them so that users can install or remove the package they want. “Apt-get” is capable of accessing package repositories using an internet connection. It calculates the package’s dependencies, downloads the required packages, and invokes the “dpkg” package management tool to install the packages.

The Application Manager visible to Maemo users is a graphical user interface to the standard Debian package management infrastructure. Using “apt-get” or “dpkg” in a terminal window of a Maemo device is possible, but relevant only for developers and advanced users.

See also the following links for further information:

Differences between Debian and Maemo packaging:

[http://wiki.maemo.org/Packaging#Differences\\_between\\_Debian.2FUbuntu\\_and\\_Maemo](http://wiki.maemo.org/Packaging#Differences_between_Debian.2FUbuntu_and_Maemo)


Information about the .service file, needed for the DBUS support:

[http://wiki.forum.nokia.com/index.php/Qt\\_application\\_for\\_Maemo\\_with\\_DBus\\_support](http://wiki.forum.nokia.com/index.php/Qt_application_for_Maemo_with_DBus_support)

## Debian Package Creation

Forum Nokia

- Qt project folders with a proper package structure and packages are made with MADDE
- Two phases in creation
  - Create a package folder containing debian package structure from a template
    - `mad pscreate -t <template name> <debian package name>`
  - Create the actual package by invoking the following command in the package folder:
    - `mad dpkg-buildpackage`
  - The package file is created to the parent folder and is of a form `<package>_<version>_armel.deb`
  - See [MADDE Qt example](#) for details



When creating package a structure for the project, it is advisable to use a wrapper folder, where the `mad pscreate` command is executed, which creates a package folder containing the package structure. This is to allow shadow-building folders for the other targets to be located nicely in the same wrapper folder as the actual folder. Further, the `dpkg-buildpackage` creates the debian package files into the parent folder of the package folder, which would be the wrapper folder.

ESbox contains a wizard for creating a set of skeleton Debian package build files, and for actually building a package. You can find the commands by selecting the project in the Project Explorer and clicking the right mouse button.

There are many files generated, some of which can be removed.

Packaging a Qt application:

[http://wiki.maemo.org/Packaging\\_a\\_Qt\\_application](http://wiki.maemo.org/Packaging_a_Qt_application)



# Control File

- Edit the maintainer field
- Define build dependencies
- Add depends info
- Add some package description
- Add build dependency for Qt library, if Qt is used (*libqt4-dev*)
- Add dependencies to proper libqt4-core, etc., packages

```
Source: myqtapplication
Section: user/other
Priority: extra
Maintainer: unknown <john.smith@nokia.com>
Build-Depends: debhelper (>= 5), libqt4-dev
Standards-Version: 3.7.2
Package: myqtapplication
Architecture: any
Depends: ${shlibs:Depends}, ${misc:Depends}
Description: <insert up to 60 chars description>
<insert long description, indented with spaces>
```



The control file contains all the dependency, conflict, and feature information of the package. Each package has a control file. Dependency, conflict, and feature information is also called rules. The rules in Debian control files are called priority, section, architecture, build-depends and depends.

- The priority rule specifies how important this package is in the installed system. The available options are essential, required, important, standard, optional, and extra. The essential option means the package is always present.
- The section rule specifies to which section the package belongs. Maemo Application Manager only shows packages which belong to some “user” subsection. Currently available subsections are: “desktop”, “development”, “education”, “games”, “graphics”, “multimedia”, “navigation”, “network”, “office”, “other”, “science”, “system” and “utilities”.
- The architecture rule specifies the target architecture, for example, i386. Using “any” is a good choice.
- The build-depends rule lists the packages from which this package depends on.
- The depends rule means that the package will not work without another package. By adding `${shlibs:Depends}`, `${misc:Depends}` the debhelper tool attempts to add direct package dependencies automatically.

## Integration to Application Framework

- In Maemo, .desktop file is required to make your Maemo application visible in the application menu
- It contains information such as application name, icon logical name, application binary file name, etc.
- Named as [app name].desktop
- During the installation, must be copied to /usr/share/applications/hildon
  - This is usually handled by adding "INSTALLS" item "desktop" to the project file (see [here](#))

```
[Desktop Entry]
Encoding=UTF-8
Version=1.0
Type=Application
Name=My Qt Application
Exec=/usr/bin/myqtapplication
Icon=myqtapplicationicon
StartupWMClass=
X-Window-Icon=myqtapplicationicon
X-HildonDesk-ShowInToolbar=true
X-Osso-Type=application/x-executable
Terminal=false
```



Some of the important fields are:

- Encoding - The character encoding of the file, should be UTF-8.
- Version - Version of the application desktop file.
- Type - Type of the entry, should be Application.
- Name - The name of the application which will come visible to task launcher menu.
- Exec - The application binary which will be started from the menu entry.
- Icon - Name of icon that represents this application. The icon will be visible in task launcher before application name. Application can install its own icon, or use one of the pre-defined icons available in platform. Format of the file should be PNG and the suffix (.png) should not be defined.

## Qt Application Deployment with MADDE

Forum Nokia

- First, create a device runtime (shorthand for the device) as follows:
  - `mad-admin create <runtime name> -a <ip-address>`
- After creating the debian package, send and install it:
  - `mad remote -r <runtime name> send ../<debian package file>`
  - `mad remote -r <runtime name> install <debian package file>`
- Finally, run the application either from the application list of the device or with:
  - `mad remote -r <runtime name> run myqtapplication`



Use `mad list` to show the available runtimes as well as targets

You can remove the device runtime as follows:

```
mad-admin remove <runtime name>
```

With “mad remote” command, you can send executable files to device, run and stop them, send package files, and install them remotely.

For example, with the following command you can ping the device:

```
mad remote -r <runtime name> ping
```

The following command gives you a shell access:

```
mad remote -r <runtime name> shell
```

See [Device runtime](#) for more information

## Scratchbox Usage – Packaging

Forum Nokia

- Now we are inside Scratchbox
- Rename your project folder (required by `dh_make`)
  - `mv myqtapplication myqtapplication-1.0`
- Create the basic package structure into a debian folder
  - `dh_make --createorig --single -e my.name@company.com -c gpl`
  - A single application package created making a copy of the original source archive and using a GPL license
- Edit rules, control, copyright, pro etc. files as needed
- Build the actual package
  - `dpkg-buildpackage -rfakeroot -b // Binary distribution`
- Look at the details how to edit the package files and the Qt .pro file:
  - [http://wiki.maemo.org/Packaging\\_a\\_Qt\\_application](http://wiki.maemo.org/Packaging_a_Qt_application)
  - [http://wiki.forum.nokia.com/index.php/Qt\\_for\\_Maemo\\_Developers\\_Guide](http://wiki.forum.nokia.com/index.php/Qt_for_Maemo_Developers_Guide)

NOKIA

One more file is needed before we can create the Debian package. This file is the control file. The control file is created by the packaging tools, but some of the fields must be edited by hand. The file contains the necessary dependencies between packages. An example control file is provided in the next slide. After editing the control file, the installation package can be created:

- Build the files by running the command `make distcheck`. It also creates a tarball for you.
- Create a new directory in the parent directory and go to that directory.
- Copy the tarball to a new directory.
- Extract the files in the tarball.
- Change directory to `hw-1.0` and make the package using the `dh_make` command.
- Answer `s` to the query to create a single binary package.
- Now there is a new Debian folder in your package folder. Find a file named `control` in that directory and edit it. You can add the package maintainer's name and e-mail address in the maintainer field. The `build-depends` field should contain the libraries you used in your application, and the `depends` field should be Maemo. You may add a free description as well.
- Symbolic links can be created during installation and removed when the package is removed. We need one symbolic link to the desktop file. Create a new `hwx.links` file into debian folder.
- Build the package with the `dpkg-buildpackage` command in the directory just above `debian`.

Now the package is ready. It can be installed using the `dpkg` command as well.

## Qt Application Deployment in the Device

Forum Nokia

- Move the application binary or Debian package to the device
  - The simplest solution to get started is to use the Maemo device as a USB mass storage, copy the binary or package to the device and unmount the device.
  - Unmounting is not needed, if the “PC Suite Mode” is used, but it needs Nokia PC Suite or Ovi Suite installed
- If using a Debian package, you need to also install it:
  - `$ sudo gainroot`
  - `$ dpkg -i <debian package file>`
- Finally, run the application:
  - `$ run-standalone.sh myqtapplication`

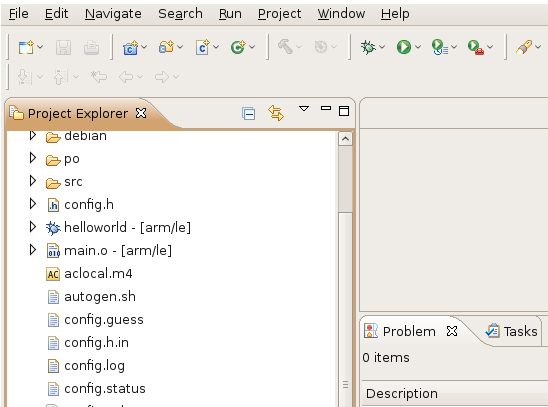


The executable inside MyDocs can not be run, so it need to be moved to, for example, `/usr/local/bin`, and the `chmod` must be invoked to allow executing

ESbox IDE

Forum Nokia

- ESbox IDE helps with Scratchbox usage
  - Eclipse using Scratchbox
- Provides
  - Documentation
  - Building (and generation of all required files)
  - X Server initialization
  - Debugging
  - Debian package creation
  - Device installation
- Start by File > New > C++ Maemo Project > Qt Hello World



NOKIA

New sample Qt Hello World project can be created from:

File > New > C++ Maemo Project > Qt for Maemo > Qt Hello World

Existing Qt project can be imported from

File > New > C++ Maemo Project > Simple > Existing Qt Program

# Deployment Using ESbox

- Create a new folder to contain package eventually
- Create the package structure
- Several files created
  - debian/rules
  - debian/control
  - debian/copyright
  - debian/compat
  - debian/changelog
- Edit the files, if you need

Options Page

Email:  
petri.niemi@digia.com

Package name:  
helloworld

Version:  
1.0

Copyright License:  
☒ No License  
☐ LGPL  
☐ BSD  
☐ GPL  
☐ Grab the whole desktop  
☐ Artistic

Package Class:  
☒ Single  
☐ Kernel  
☐ Cdb  
☐ Multiple  
☐ Library

☐ Native  
☐ Add missing  
☐ Defaultless

Finish

Cancel

NOKIA

Forum Nokia

# Deployment Using ESbox

- After creation of the package structure, just create the package, e.g., from the context menu

Debian Package Reader

Built successfully - helloworld\_1.0-1\_armel.deb

Package	Size	Location
changelog.Debian.gz	195	/usr/share/doc/helloworld
com.nokia.helloworld.service	67	/usr/share/dbus-1/services
copyright	621	/usr/share/doc/helloworld
helloworld	6668	/usr/bin
helloworld.desktop	294	/usr/share/applications/hildon
README.Debian	183	/usr/share/doc/helloworld

?

OK

File Edit View Terminal Tabs Help

training@training12:/scratchbox/users/training/home/training/workspace/packaging\$ ls -l

total 196

-rw-r--r-- 1 training training 688 2009-10-16 06:46 helloworld\_1.0-1\_armel.changes

-rw-r--r-- 1 training training 5510 2009-10-16 06:46 helloworld\_1.0-1\_armel.deb

-rw-r--r-- 1 training training 272 2009-10-16 06:46 helloworld\_1.0-1.dsc

-rw-r--r-- 1 training training 178304 2009-10-16 06:46 helloworld\_1.0-1.tar.gz

training@training12:/scratchbox/users/training/home/training/workspace/packaging\$

NOKIA