Archived:Getting started with MADDE on Windows

Archived: This article is archived because it is not considered relevant for third-party developers creating commercial solutions today. If you think this article is still relevant, let us know by adding the template {{ReviewForRemovalFromArchive|user=~~~~|*write your reason here*}}.

You can now get started with Qt development for Maemo more easily with the Qt SDK. Using the Qt SDK you have one simple installation and don't need to install the Maemo 5 SDK. For more information, visit the Tools page on Nokia Developer and read the Getting Started Guide.
Notes:

- MADDE was used during Qt on Maemo technology preview stages but is not recommended because the Qt SDK now provides an easier path for building for Maemo (device/simulator)
- MADDE is integrated into the Qt SDK - see C:\QtSDK\Maemo\4.6.2
- The document is up to date on 29 April 2011

Introduction

11 Apr 2010

MADDE (**M**aemo **A**pplication **D**evelopment and **D**ebugging **E**nvironment) is a technical development tool for Maemo. It can be used on Windows, Linux (32-bit/64-bit) and Mac OS. This article is a quick start guide to MADDE development on Windows platform. After reading (and understanding) this article, you should be able to setup MADDE on Windows, create a simple Qt project, compile and build the Qt project for Maemo (Fremantle), deploy and run it on an Nokia N900 which is based on Maemo 5 (Fremantle).

In case, you are interested in MADDE for other supported platforms than Windows (i.e. Linux (32-bit/64-bit) and Mac OS X), please follow the links in the Guides section here.

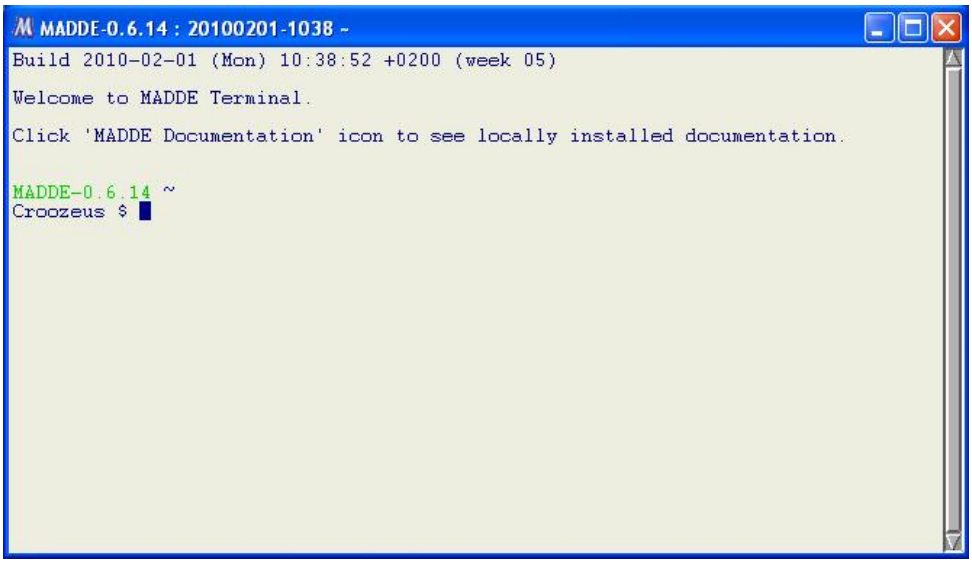
Download and Install

1. MADDE for Windows (install-madde-0.6.14.exe) is 642 MB large file that you need to download from this link. You need to read and accept the End User Software Agreement before downloading.
2. Run the install-madde-0.6.14.exe file after downloading and complete the installation with the default options. If you did so, you should have MADDE installed in path C:\MADDE (provided you have windows on C:) Two shortcuts are placed on the desktop, viz. MADDE Terminal and MADDE Documentation; out of which MADDE Documentation shortcut is broken. (Note, MADDE is a technical preview package)

Note: Please check Archived:Getting started with MADDE on Windows#Tips for using Qt 4.6.2

Configuration

Launch the MADDE Terminal by using the shortcut on the desktop. You should see a command prompt, illustrated in the screenshot below.



To check the targets and runtime which are available to MADDE command line, use the following command.

```
mad list
```


(C) Copyright Nokia 2013. All rights reserved.

```
MADDE-0.6.14 : 20100201-1038 ~

MADDE-0.6.14 ~
Croozeus $ mad list

Targets:
fremantle-qt-0951  (default)
maemo412-1         (installable)

Runtimes:

MADDE-0.6.14 ~
Croozeus $
```

If you have more than one targets/runtimes installed, you can select a default target/runtime by using the following command

```
mad set <target name>
```

For now, we do not need to do anything since we have only fremantle-qt-0942 target installed and set as default already.

Create, Compile and Build a quick project

Now that we have the environment ready on the PC, we can create a Qt project. Alike various development environments, MADDE provides some templates as the skeletons for the applications. To see the available templates, use the following command.

```
mad pscreate -l
```

```
MADDE-0.6.14 : 20100201-1038 ~

MADDE-0.6.14 ~
Croozeus $ mad pscreate -l
TEMPLATE NAME    TYPE     DESCRIPTION
empty            prog     Empty project
lib_simple       lib      Simple example for C/C++ libraries
qt_simple        qtprog   Simple example for QT programs
simple           prog     Simple example for C/C++ programs

MADDE-0.6.14 ~
Croozeus $
```

Since we are creating our first project, lets begin by creating a skeleton of qt_simple. To do so, use the following command.

```
mad pscreate -t qt_simple qthello
```

where `qt_hello` is the name of the project. This would create a skeleton for our project. List the directories to see the skeleton, as shown in the screenshot.

```
MADDE-0.6.14 : 20100201-1038 ~/qthello/src

MADDE-0.6.14 ~
Croozeus $ cd qthello

MADDE-0.6.14 ~/qthello
Croozeus $ ls
Makefile  build  build-stamp  configure-stamp  data  debian  qtprog.pro  src  uis

MADDE-0.6.14 ~/qthello
Croozeus $ cd src/

MADDE-0.6.14 ~/qthello/src
Croozeus $ ls
qtmain.cpp

MADDE-0.6.14 ~/qthello/src
Croozeus $ cat qtmain.cpp
/*
 * Created: 03/09/10-20:03:24
 * Author: Croozeus
 */
#include <QApplication>
#include <QPushButton>

int main(int argc, char *argv[])
{
        QApplication app(argc, argv);

        QPushButton hello("Hello world!");

        hello.show();
        return app.exec();
}

MADDE-0.6.14 ~/qthello/src
Croozeus $
```


(C) Copyright Nokia 2013. All rights reserved.

You may view or edit a file by using various linux commands, for example

```
cat qtmain.cpp and vi qtmain.cpp
```

If you like to change anything, please feel free to do so otherwise don't bother to edit the code, since its your first application.

To compile the project use the following commands from the project directory,

```
mad qmake
mad make
```

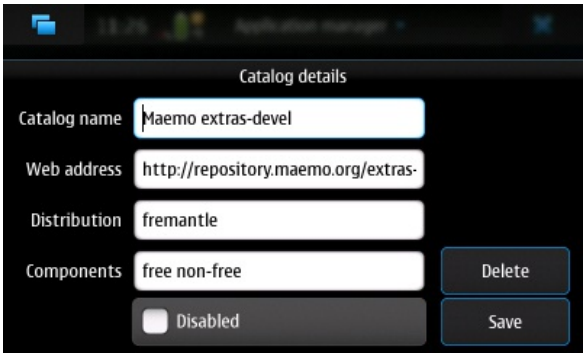
MADDE applications can only be tested on a real device (can only build for arm) so we create a debian package by using the following command.

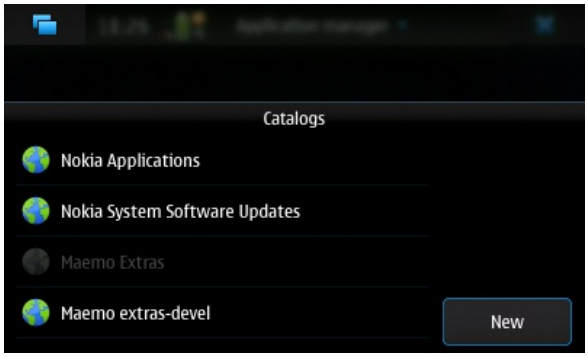
```
mad dpkg-buildpackage
```

This would create a debian package `qthello_0.1_armel.deb` in the parent directory. So our sample application is ready to be tested on a Nokia 900 device.

Configuring Nokia N900 with MADDE

For configuring the Nokia N900 with MADDE, a Mad-developer application is required to installed on the N900 device. Mad-developer application is available in the *extras-devel* repository. If you don't have the *extras-devel* catalog in your application manager, you can add it manually, as illustrated in the screenshots below. (Web address: http://repository.maemo.org/extras-devel//)

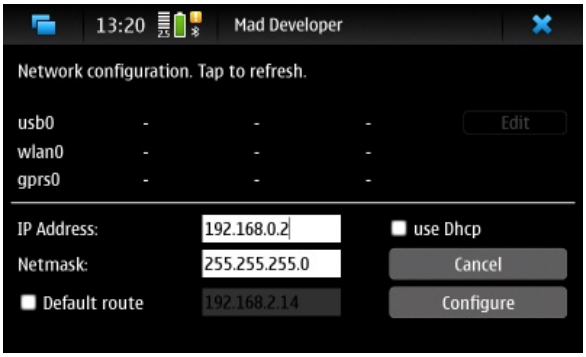




Note: Optional: Besides Mad-developer, you need to also install rootsh.install available in the extras-devel also. This is required to gain the root access from the Mad-devel application (used to set usb0 from the x-terminal). So, again install rootsh from the application manager.

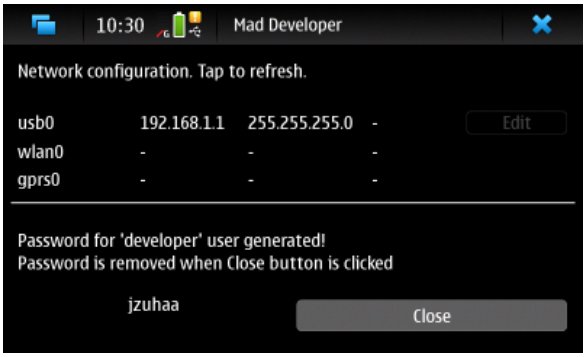
Launch the Mad-developer application on the device. It has 3 network configuration options, viz. usb, wifi and gprs. We would configure a usb configuration here, the other two configurations can be done similarly too.

Edit the usb0 configuration, assign an IP address to it and click on Configure. We have assigned 192.168.0.2 as shown in the screenshot below.



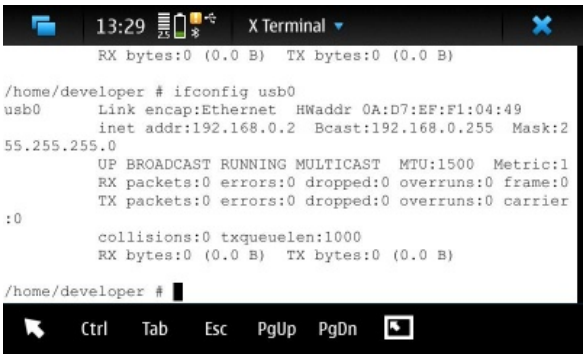
So that being done, the usb0 connection of the device is configured to 192.168.0.2 as shown in the screenshot below.

(C) Copyright Nokia 2013. All rights reserved.



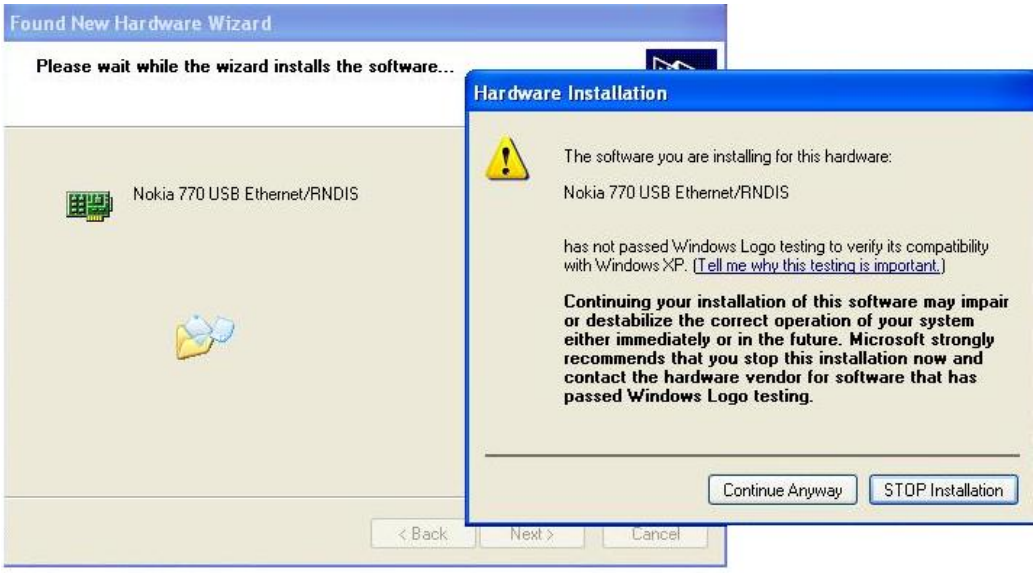
Note: You can also view the network configurations by using the X-terminal.

```
ifconfig
```



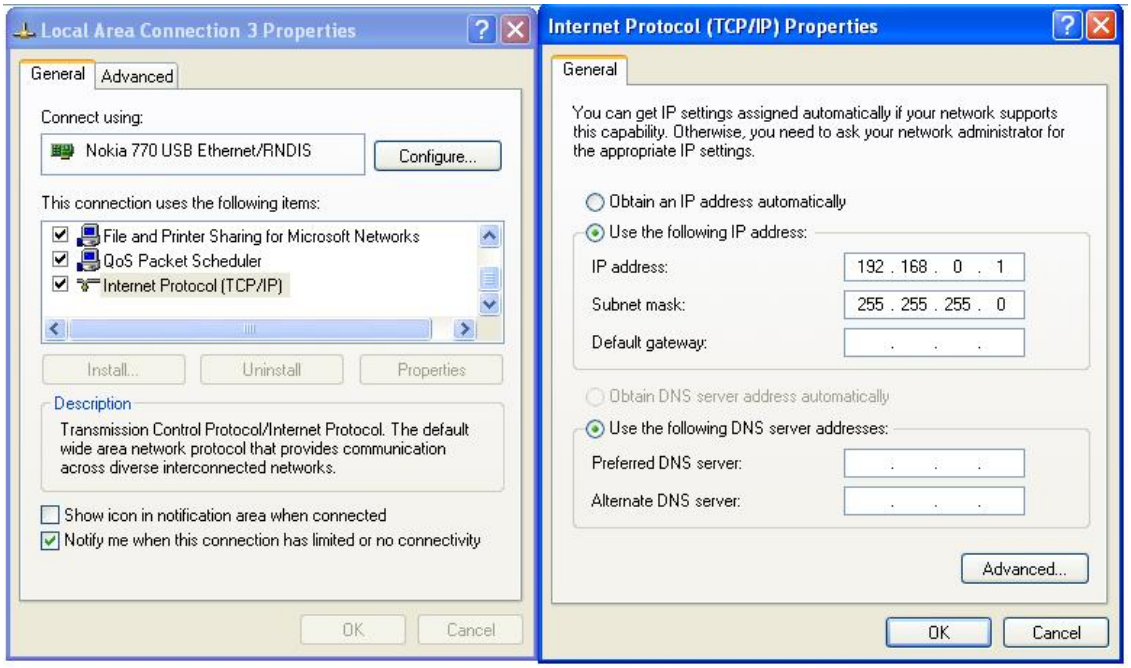
Connect the Nokia N900 to the PC via the usb cable and select **Manage usb->Load g_ether** in the Mad developer application. When prompted on connection, choose connection via PC suite.

You would need to install a driver for the N900. Windows will ask you to name a device driver for the RNDIS gadget. Download the "Nokia770 USB Ethernet RNDIS.inf" from here and install it as a driver.

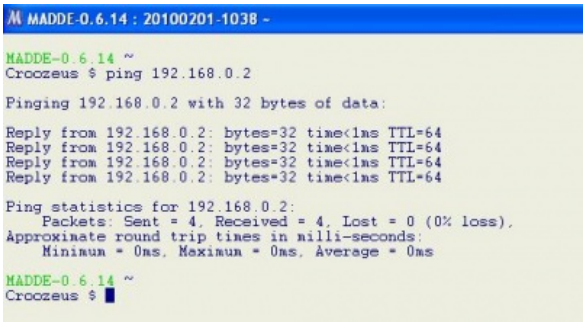


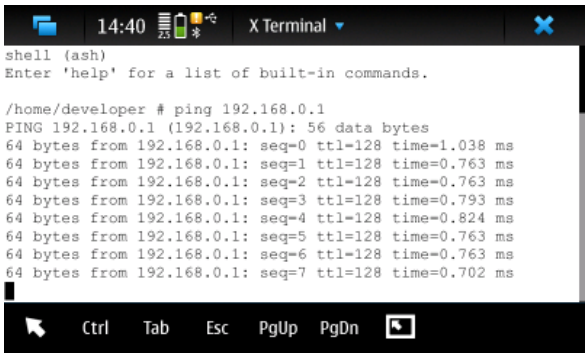
A Local Area Network connection would be established with the N900, once this driver is installed. Assign another IP in the with the same Netmask to the PC. We have assigned 192.168.0.1 as shown in the screenshot below.

(C) Copyright Nokia 2013. All rights reserved.



The connection between the PC and the N900 is established. This can be checked by pinging each other from each other, as illustrated below.





Create a runtime for N900 on MADDE console by the following command.

```
mad-admin create N900 -a 192.168.0.2
```

You would be prompted for a password. Choose Developer Password from the Mad-developer application on N900 to get the password. Provide this password to the MADDE console. This should successfully set up the device runtime. Now the runtime is listed as a runtime, when you run

```
mad list
```

```
MADDE-0.6.14 ~
Croozeus $ mad-admin create N900 -a 192.168.0.2
Device runtime 'N900' initialized!

MADDE-0.6.14 ~
Croozeus $ mad list

Targets:
fremantle-qt-0951   (default)
maemo412-1          (installable)

Runtimes:
N900  (installed)

MADDE-0.6.14 ~
Croozeus $
```

Testing the application on Nokia N900

To check if the N900 runtime is available and up, use the following command.

```
mad remote -r N900 ping
```

If you get “Runtime up and connected” you may proceed to test your application. If not, please check the device and PC IP addresses.

Send the debian package by executing the following command in the project directory.

```
mad remote -r N900 send ../qthello_0.1_armel.deb
```

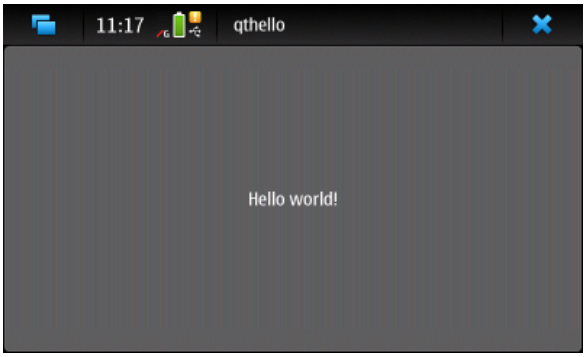

(C) Copyright Nokia 2013. All rights reserved.

Install the package by

```
mad remote –r N900 install qthello_0.1_armel.deb
```

Launch the application in the device by the following command,

```
mad remote –r N900 run qthello
```



Terminate the application in the device by the following command.

```
mad remote –r N900 stop qthello
```

To uninstall a debian package, use the following command.

```
mad remote –r N900 uninstall qthello
```

Tips

The above sections mentioned the necessary steps which you are required to execute - to code, compile, package and deploy any Qt applications on Nokia N900 using MADDE. Here are a few tips that might speed up your development on MADDE.

1. If you are keen on using a versatile IDE for creating your projects, you can configure MADDE with Qt Creator on windows. Note that, MADDE is not able to be configured with the latest version of Qt Creator. The version has to be >= 1.3.80. Find instructions on configuring MADDE with Qt Creator here
2. Navigate to the MADDE installation directory/home/user (C:\MADDE\0.6.14\home\User) you will find here the projects you created using MADDE. For example navigate to the sample application C:\MADDE\0.6.14\home\Croozeus\qthello\src, you will find the qtmain.cpp. Open this file with a text editor of your choice edit it, save it. Add more files to the src folder if needed. Compile, build, package using the MADDE environment. Better than using a linux command prompt editor
3. Once your target runtime (N900 in this article) is created on MADDE, you can acess the shell of the device and also gain root access to it. Note: You need to have installed “Rootsh” for gaining the root as described in the article earlier.

   ```
   mad remote -r N900 shell
   sudo gainroot
   ```

4. Should you wish to test the application without making a debian package, use the following command directly from the project parent folder.

   ```
   mad remote -r N900 send build/qthello
   mad remote -r N900 run qthello
   ```

5. **Qt 4.6.2 for Maemo 5** has been released. While MADDE is not available for it as of now, an experimental version of MADDE with support for Linux, Mac OS X and Windows can be found here. For windows, you need to download qt4-maemo5-windows-20100212.tar.gz and execute the following commands using the MADDE terminal.

   ```
   cd /sysroots/fremantle-arm-sysroot-2.2009-51-1-qt453
   mkdir opt
   cd opt
   tar -xzvf /qt4-maemo5-windows-20100210.tar.gz
   ```

   To build a project you can use the qmake and make as shown below

   ```
   mad /sysroots/fremantle-arm-sysroot-2.2009-51-1-qt453/opt/qt4-maemo5/bin/qmake
   mad make
   ```

6. If you wish to use MADDE on Windows 7, follow instructions at How to make MADDE work on Windows 7 64-bit.

Author

This article is written by croozeus 21:28, 10 March 2010 (UTC). Comments and improvements regarding the content of this article are welcome.

(C) Copyright Nokia 2013. All rights reserved.

(C) Copyright Nokia 2013. All rights reserved.