

Quick guide how to create .deb package on your N9/50

1. Go to folder **/home/user/MyDocs/.DEBCREATOR** (you will see this folder after installation)
2. Create here folder with name of your package, for example **my-package**
3. Inside the **my-package** folder, create folder named **DEBIAN**
4. Now you got: **/home/user/MyDocs/.DEBCREATOR/my-package/DEBIAN**
5. Inside the **DEBIAN** folder you need put a few files:
 - **control** (needed) and **preinst, postinst, prerm, postrm** (if you want/need)
6. Example how the basic control file should look (all other stuff can be added automatically by application):

```
Package: my-package
Version: 0.0.1
Architecture: armel
Maintainer: Schturman <schturman@blabla.com>
Installed-Size: 260
Section: user/other
Priority: optional
Homepage: http://www.your-site.com
Description: Deb packages creator for harmattan (N9 & N950)...
Maemo-Display-Name: DEB packager
Maemo-Flags: visible
```

You can create this file by yourself or app can create it for you. If you start app without **control** file inside the DEBIAN folder, the app start to create it and ask you some questions. If you skip all questions by pressing enter, it will insert default parameters like in this example (red text):

```
Create control file for "big snooze2" package
```

```
Insert version number:
```

```
Insert Maintainer:
```

```
Insert Section:
```

```
Insert homepage:
```

```
Insert Description:
```

```
Insert Display-name:
```

```
Control file created, check it...
```

Package: bigsnooze2
Version: 0.0.1
Architecture: armel
Maintainer: It's me...
Installed-Size: 100
Section: user/other
Priority: optional
Homepage: <http://talk.maemo.org/showthread.php?t=91053>
Description: My first package created via N9 Packages creator.
Maemo-Display-Name: My package
Maemo-Flags: visible

Press C to change or Enter to continue:

You can press “C” to start create it again (fix) or enter to continue creating package...
Example with parameters that I inserted (red text):

Create control file for "bigsnooze2" package

Insert version number: 2.3.5

Insert Maintainer: Schturman <schturman@mail.com>

Insert Section:

Insert homepage: <http://www.my-site.com>

Insert Description: This test package for PDF instruction. I learn how to create packages via N9 Packages creator.

Insert Display-name: Big Snooze2

Control file created, check it...

Package: bigsnooze2
Version: 2.3.5
Architecture: armel
Maintainer: Schturman <schturman@mail.com>
Installed-Size: 100
Section: user/other
Priority: optional
Homepage: <http://www.my-site.com>
Description: This test package for PDF instruction. I learn how to create packages via N9 Packages creator.
Maemo-Display-Name: Big Snooze2
Maemo-Flags: visible

Press C to change or Enter to continue:

If you don't know which “Section” you need, ALWAYS SKIP this question by pressing Enter and it will insert default parameter: user/other !!!

7. Add icon to your **control** file if you want to see icon in app manager and through installation. You have two methods to do this:

Method 1:

Application will do it automatically for you. Just put ONE icon in **.png** format to your package

folder: **/home/user/MyDocs/.DEBCREATOR/my-package**. Name of icon is not important, dimension as well is not important, you can put 90x90, 80x80, 64x64 etc.. The application will resize it (if needed) to 64x64 and insert to **control** file.

Method 2:

Manual method...

1. Create icon with dimension **64x64px** and extensions **.png**, for example **my-package64.png**
2. Put this icon, for example to: **/home/user/MyDocs/my-package64.png**
3. Open Terminal and run this 2 commands:

```
echo Maemo-Icon-26: >> /home/user/MyDocs/.DEBCREATOR/my-package/DEBIAN/control  
base64 /home/user/MyDocs/my-package64.png | sed 's/^/ /' >> /home/user/MyDocs/.DEBCREATOR/my-package/DEBIAN/control
```

Now your control file looks like this:

```
Package: my-package  
Version: 0.0.1  
Architecture: armel  
Maintainer: Schturman <schturman@blabla.com>  
Installed-Size: 260  
Section: user/other  
Priority: optional  
Homepage: http://www.your-site.com  
Description: Deb packages creator for harmattan (N9 & N950)...  
Maemo-Display-Name: DEB packager  
Maemo-Flags: visible  
Maemo-Icon-26:  
iVBORw0KGgoAAAANSUHEUgAAAEAAAABACAYAAACqaXHeAAAKRWIDQ1BJQ0M  
U2dUU+kWPffe9EJLiICUS29SFQggUkKLgBSRJiohCRBKicGh2RVRwRFFRQQbyKCIA4
```

8. Now go to: **/home/user/MyDocs/.DEBCREATOR/my-package** and put here your moded files with full path where you want to replace them. For example: **/opt/my-package/my-script.sh**
Or for example you want to replace stock browser icon located in:
/usr/share/themes/blanco/meegotouch/icons/icon-l-browser.png Just create here a full path to this icon or manually or with command:

```
mkdir -p /home/user/MyDocs/.DEBCREATOR/my-package/usr/share/themes/blanco/meegotouch/icons
```

And put here your new icon **icon-l-browser.png**.

9. Another way, you can use **postinst** script to copy your file to destination.
10. **Don't forget to create backup** of the files that you go to replace. For this you can use **preinst** script and for restore original one, use **prerm** or **postrm** scripts for example.
11. Also, if you want to add **aegis-manifest** file, you can do it too, just create file named:
my-package.aegis and put this file to your package folder:
/home/user/MyDocs/.DEBCREATOR/my-package
12. The content of **my-package.aegis** should look in this format:

```
<aegis>
  <request policy="add">
    <credential name="UID::root" />
    <credential name="GID::root" />
    <credential name="GRP::root" />

    <for path="/opt/my-package/my-script.sh" />
  </request>
</aegis>
```

This example if you want to run your script as **ROOT** from the desktop icon that should look like this:

```
[Desktop Entry]
Type=Application
Name=My App
Categories=System;
Exec=/usr/bin/meego-terminal -n -e /opt/my-package/my-script.sh
Icon=/usr/share/icons/hicolor/80x80/apps/my-package80.png
```

You need give a right permission to your executable file (**my-script.sh**). You can do it by just adding this line in **postinst** script:

```
chmod 4755 /opt/my-package/my-script.sh
```

Or you can use **rules** file (see below) that include commands that give a right permission to your executable file **before packaging**.

Now add Manifest line to your control file too: **Aegis-Manifest: included**

Example:

```
Package: my-package
Version: 0.0.1
Architecture: armel
Maintainer: Schturman <schturman@blabla.com>
Installed-Size: 260
Section: user/other
Priority: optional
Homepage: http://www.your-site.com
Description: Deb packages creator for harmattan (N9 & N950)...
Aegis-Manifest: included
Maemo-Display-Name: DEB packager
Maemo-Flags: visible
Maemo-Icon-26:
iVBORw0KGgoAAAANSUgAAAEAAAABACAYAAACqaXHeAAAKRWIDQ1BJQ0M
U2dUU+kWPffe9EJLiICUS29SFQggUkKLgBSRJiohCRBKICGh2RVRwRFFRQQbyKCIA4
```

If you forgot to add this line to the **control** file, it also okay. Application will add it automatically for you.

13. Creating **rules** file. Just create file named **rules** that include permission commands to your files. **DON'T create this file as script!** Put created file **rules** to your package folder:

/home/user/MyDocs/.DEBCREATOR/my-package

See example:

```
chmod 4755 /opt/DEBCREATOR/my-package/opt/my-package/test-script.sh
chmod 777 /opt/DEBCREATOR/my-package/opt/my-package/tar
```

Or like this:

```
cd /opt/DEBCREATOR/my-package/opt/my-package
chmod 4755 test-script.sh
chmod 777 tar
```

REMEMBER!:

- You always need to write **/opt/DEBCREATOR/my-package** before a real path to your file (**/opt/my-package/test-script.sh**).

14. In this folder **/home/user/MyDocs/.DEBCREATOR** you can put a quantity of folders (your packages) as much as you want, but the app can show **ONLY first 5** folders ! You can hide folders from app that you not use at this time by adding dot at the beginning of the name (**.my-package**), especially if this folders a first 5 folders...

15. Also for test and learning you can download this [iconchanger.tgz](#) and put it to: **/home/user/MyDocs**

- Open terminal as **ROOT** and write:

```
tar xvpfz /home/user/MyDocs/iconchanger.tgz -C /home/user/MyDocs/.DEBCREATOR
```

- It will extract all files of the demo Icon changer package and it ready to packaging. Just look into the files how I did it...
- Also you can download this demo package ready to install and try it, see how it work after installation: [iconchanger.deb](#). This package will replace icon of empty folder to new one and add app icon to desktop that you can check how work script as root from desktop icon.

16. Now, run “**DEB packager**” app from your desktop, then choose numbers of packages that you want to create separated by one space (1 4 5...). It will create your **.deb** packages and you will find them in their folders: **/home/user/MyDocs/.DEBCREATOR/my-package**

Inference:

After all this big text, you can do it in 5 simple steps:

1. Put your files in package folder.
2. Create scripts **preinst**, **postinst**, **prerm**, **postrm** (if you want/need) in DEBIAN folder.
3. Create **rules** and **aegis-manifest** files (if you need) in package folder.
4. Put **icon** in package folder for adding to **control** file.
5. Run **N9 Packages creator** and get your application 😊

PS. DEB packager have dependencies: [imagemagick](#), [libmagick10](#)

Both packages will be installed automatically if you install it by double clicking on the file. If you install this package from terminal (as ROOT) via **dpkg -i** , run this command at the end also as ROOT (internet connection required in both variations):

```
apt-get install -f -y
```

Also, you can add my repo for updates by running this 3 commands as ROOT (or from N9QT -> tweak II):

```
echo 'deb http://harmattan.openrepos.net/Schturman personal main' >  
/etc/apt/sources.list.d/openrepos-Schturman.list
```

```
wget -O - http://harmattan.openrepos.net/openrepos.key | apt-key add -
```

```
apt-get update
```

And you will get system updates when update will be ready ☺

**Use created .deb packages on your own risk !
I'm not responsible for any damage if it will cause to your phone ;)
Good luck...**

Thanks for using my script,
Schturman