

Nokia N9 For Linux Users

Since about 20% of our lab are N9 users, it seems worth maintaining a page for sharing tips and tricks.

Synchronizing Contacts and Calendars

If you do not want to entrust your data to *The Cloud*, you can sync the N9 directly to the desktop. Here is the most functional way I found:

- On the N9, install [syncevolution*_armel.deb](#).
- On the host PC, run `syncevo-http-server -q http://localhost:9000/syncevolution`.

This allows synchronization of the contacts and of multiple calendars. Todos and notes cannot currently be sync'ed in this way, but this may [follow](#) at some point.

There does not currently seem to be a straightforward way for SyncEvolution to sync over [Bluetooth/OBEX](#). One can use the N9 built-in SyncML client to sync with a desktop Evolution server over Bluetooth, but this solutions apparently has its [weaknesses](#).

I do not miss Bluetooth sync much, as I need to keep the USB cable around for recharging anyway.

Synchronizing Notes

Notes can be sync'ed both ways as follows:

- On both the N9 and on the host system, install `git`.
- Place all your notes in flat files, one file per note.
- On the N9, place a couple of scripts that `git push` and `git pull` your Notes directory.
- A full sync is then done by something like the following sequence:
 1. `push` on the N9
 2. `pull` on the host
 3. `push` on the host
 4. `pull` on the N9

Voilà. Much better than conventional solutions, since notes are sync'able even if they are simultaneously edited on both sides. And you get to keep your entire history.

The drawback is lack of integration with the N9. Editing notes can be tedious. You can alleviate this a bit by [configuring](#) the N9 to open text files inside nano instead of the document viewer.

Synchronizing Encrypted Data

Many people carry encrypted data around with them, such as passwords. There are various *keyring*,

wallet etc. programs out there to maintain these, but I did not find a good solution for the N9, at least none that allowed any straightforward way of sync'ing such data with the desktop. But here is a simple solution that covers most use cases:

- On both the N9 and on the host system, install `gpg`.
- Write little scripts for encryption and decryption of files.
- Keep all your encrypted data in a subtree of the file system, one record per (encrypted) file.
- Sync this directory subtree from the desktop to the N9 using `rsync`.
- [Configure](#) the N9 to associate with `.gpg` files the action of running, inside the Meego Terminal, a script that decrypts it to `stdout`.

Voilà. Much better than any keyring/wallet program, at least for Emacs users. To consult a password on the N9, simply open the file in a file manager (I recommend Filebox), and it will pop up the terminal, ask for your passphrase, and dump the record to the screen.

The drawback is that encrypted files can only be edited on one of the two machines, not on both.

Offline Reading

I have a little script that pulls a URL with prerequisites using `wget`, cleans the pulled HTML files with `tidy`, and strips out all `<script>`s with a tiny XSLT transformation (I do not know of a way to turn off Javascript with the stock Web browser). The whole batch is then moved to the N9 by my sync script, and I always have a bunch of online reading material with me.

From:

<https://iis.uibk.ac.at/> - IIS

Permanent link:

<https://iis.uibk.ac.at/misc/n9?rev=1327700222>

Last update: **2018/09/03 14:57**

