

# 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, Calendars, To-Dos

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 the latest [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 and to-do lists.<sup>1)</sup> Notes cannot currently be sync'ed in this way, but this [is in the works](#).

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

## Other Options

- 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 solution apparently has its [weaknesses](#).
- OpenSync is another promising route ([SyncML](#) facing the N9, [Akonadi](#) or [Evolution](#) on the host). Presumably this would be subject to the same alleged limitations of the N9's built-in Bluetooth sync as noted above. We have not tried this comprehensively (partially because of package dependency problems; OpenSync is under heavy development), and would be delighted to learn more.
- There are several options for installing your own, open-source *cloud* services locally, and sync with them, but this adds a third, intermediary data store.

## Synchronizing Notes

Notes stored in plain text files can be sync'ed both ways as follows:

- On both the N9 and on the host system, install git. On the N9, create a bare git repository.<sup>2)</sup>
- Place all your notes in flat files, one file per note.
- On the N9, place a couple of scripts that `git push`<sup>3)</sup> and `git pull`<sup>4)</sup> 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<sup>5)</sup> the N9 to open text files with an editor instead of the document viewer.

**Warning:** I have not yet tried what happens in case of a merge conflict. In any case, they will require manual fixing.

## Sensitive Data

Many people carry sensitive data around with them, such as passwords. There are various *keyring*, *wallet* etc. programs out there to maintain these in an encrypted fashion, 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.
- 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 decrypting to stdout inside the Meego Terminal.<sup>6)</sup>

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](#), and it will pop up the terminal, ask for your passphrase, and dump the record to the screen.

Drawbacks:

- Encrypted files can only be edited on one of the two machines, not on both.
- This solution is much more **vulnerable to local malware attacks** than well-written keyring/wallet programs.

## 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 offline reading material with me.

<sup>1)</sup>  
Contacts sync appears to require a login shell on the N9. Thus, my sync script fires it up with

```
ssh user@192.168.2.15 "sh -l -c 'syncevolution -q client-for-laptop
addressbook'"
```

<sup>2)</sup>  
The need for git on the N9 can be avoided entirely by sshfs-mounting the N9 and doing all of the following on the host system.

<sup>3)</sup>

```
PATH=$PATH:/usr/libexec/git-core
cd $1
git status | grep 'nothing to commit (working directory clean)'
if [ $? != 0 ] ; then
    git add -A
    git commit -m "sync commit"
    git push
fi
```

4)

```
PATH=$PATH:/usr/libexec/git-core
cd $1
git pull
```

5)

To /home/user/.local/share/applications/defaults.list, to the [Default Applications] section, add the line

```
text/plain=nano.desktop
```

or create this if it does not exist.

6)

Here is my /usr/share/applications/decrypt.desktop on the N9:

```
[Desktop Entry]
Name=Decrypt
Comment=Display GPG-encrypted text/plain files
Exec=/usr/bin/meego-terminal -e sh -c "/usr/bin/gpg --decrypt %U ; read"
Type=Application
MimeType=application/pgp-encrypted;
```

From:

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



Permanent link:

<https://iis.uibk.ac.at/misc/n9?rev=1346505334>

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