

OpenSync

synching on the free desktop

"It's the first time I feel well when thinking about synching."
- Cornelius

This interview intends to get you some insight into OpenSync, the upcoming free unified synching solution for the free desktop.

It was done and formatted by Arne Babenhauerheide in April 2006 via e-mail and KOffice.

> Hi Cornelius, Armin and Tobias,



Left to right:
Cornelius, Armin, Tobias

As you are now getting close to version 1.0 of OpenSync, which is expected to become the new synchronisation framework for KDE and other free desktops, we are quite interested in the merits it can provide for KDE users and for developers, as well as for the Open Source Community as a whole. So there's one key-question before I move deeper into the details of OpenSync:

1. Basics

> *What does OpenSync accomplish, that noone did before?*

Cornelius:

First of all it does its job of synchronizing data like addressbooks and calendars between desktop applications and mobile devices like PDAs and cell phones.

But the new thing about OpenSync is that it isn't tied to a particular device or a specific platform. It provides an extensible and modular framework which is easy to adopt for application developers and people implementing support for synching with mobile devices.

OpenSync is also independent of the desktop platform. It will be the common syncing backend for at least KDE and GNOME and other projects are likely to join. That means that the free desktop will have one common syncing solution. This is something really new.



> *How do the end-users profit from using synching solutions which interface with OpenSync as a framework?*

Cornelius:

First, the users will be able to actually synchronize all their data. By using one common framework there won't be any "missing links", where one application can sync one set of devices and another application a different one. With OpenSync all applications can sync all devices.

Second, the users will get a consistent and common user interface for syncing across all applications and devices. This will be much simpler to use than the

current incoherent collection of syncing programs you need if you have more than the very basic needs.

> How does OpenSync help developers with coding?

Cornelius:

It's a very flexible and well-designed framework which makes it quite easy for developers to add support for new devices and new types of data. It's also very easy to add support for OpenSync to applications.

The big achievement of OpenSync is that it hides all the gory details of syncing from the developers who work on applications and device support. That makes it possible for the developers to concentrate on their area of expertise without having to care what's going on behind the scenes.

I have written quite a lot of synchronization code in the past. Trust me, it's much better, if someone just takes care of it for you, and that's what OpenSync does.

Tobias:

Another point to mentioned is the python wrapper for opensync, so you are not bound to C or C++, but can develop plugins in a high level scripting language.

> Why should producers of portable devices get involved with your team?

Cornelius:

OpenSync will be the one common syncing solution for the free desktop. That means there is a single point of contact for device manufacturers who want to add support for their devices. That's much more feasible than addressing all the different applications and solutions we had before. With OpenSync it hopefully will become interesting for manufacturers to officially support Linux for their devices.

> Do you also plan to support applications of OpenSync in proprietary systems like OSX and Windows?

Cornelius:

OpenSync is designed to be cross-platform, so it is able to run on other systems like Windows. How well this works is always a question of people actually using and developing for this system. As far as I know there isn't a real Windows community around OpenSync yet. But the technical foundation is there, so if there is somebody interested in working on a unified syncing solution on Windows, everybody is welcome to join the project.

> What does your synchronisation framework do for KDE and for KitchenSync in particular?

Cornelius:

OpenSync replaces the KDE-specific synchronization frameworks we had before. Even in KDE we had several separate syncing implementations and with OpenSync we can get replace them with a common framework. We had a more generic syncing solution in KDE under development. This was quite similar from a design point of view to OpenSync, but it never got to the level of maturity we would have needed, because of lack of resources. As OpenSync fills this gap we are happy to be able to remove our old code and now concentrate on our core business.

2. Who, How and Why?

> *What was your personal reason for getting involved with OpenSync?*



Cornelius:

I wrote a lot of synchronization code in the past, which mainly came from the time where I was maintaining KOrganizer and working on KAddressBook. But this always was driven by necessity and not passion. I wanted to have all my calendar and contact data in one place, but my main objective was to work on the applications and user interfaces handling the data and not on the underlying code synchronizing the data.

So when the OpenSync project was created I was very interested. At GUADEC in Stuttgart I met with Armin, the maintainer of OpenSync, and we talked about integrating OpenSync with KDE. Everything seemed to fit together quite well, so at Linuxtag the same year we had another meeting with some more KDE people. In the end we agreed to go with OpenSync and a couple of weeks later we met again in Nuernberg for three days of hacking and created the KDE frontend for OpenSync. In retrospect it was a very pleasant and straightforward process to get where we are now.



Armin:

My reason to get involved (or better to start) OpenSync was my involvement with its predecessor Multisync. I am working as a system administrator for a small consulting company and so I saw some problems when trying to find a synchronization solution for Linux.

At that point I joined the Multisync project to implement some plugins that I thought would be nice to have. After some time I became the maintainer of the project. But I was unhappy with some technical aspects of the project, especially the tight coupling between the syncing logic and the GUI, its dependencies on GNOME libraries and its lack of flexibility.



Tobias:

Well, I have been a KDE PIM developer for several years now, so there was no way around getting in touch with synchronization and KitchenSync. Although I liked the idea of KitchenSync, I hated the code and the user interface [...]. So when we discussed to switch to OpenSync and reimplementing the user interface, I volunteered immediately.

> *Can you tell us a bit about your further plans and ideas?*

Cornelius:

The next thing will be the 1.0 release of OpenSync. We will release KitchenSync as frontend in parallel.

Amin:

There are of course a lot of things on my todo and my wishlist for opensync. For the near future the most important step is the 1.0 release, of course, where we still have some missing features in OpenSync as well as in the plugins.

One thing I would really like to see is a thunderbird plugin for OpenSync. I use thunderbird personally and would really like to keep my contacts up to date with my cellular, but I was not yet able to find the time to implement it.

Tobias:

One thing that would really rock in future versions of OpenSync is an automatic hardware detection mechanism, so when you plugin your Palm or switch on your bluetooth device, OpenSync will create a synchronization group automatically and ask the user to start syncing. To bring OpenSync to the level of *The Syncing Solution* [tm]_ we must reduce the necessary configuration to a minimum.

> *What was the most dire problem you had to face when creating OpenSync and how did you face it?*

Cornelius:

Fortunately the problems which I personally would consider to be dire are solved by the implementation of OpenSync which is well hidden from the outside world and [they are] an area I didn't work on ;-)

Amin:

I guess that I am the right person to answer this question then :)

The most complicated part of OpenSync is definitely the format conversion which is responsible for converting the format of one device to the format that another device understands.

There are a lot of subsystems in this format conversion which makes it so complex, like conversion path searching, comparing items, detection of mime types and last but not least the conversion itself. So this was a hard piece of work.

> *What was the greatest moment for you?*

Cornelius:

I think the greatest moment was when, after three days of concentrated hacking, we had a first working version of the KDE frontend for OpenSync. This was at meeting at the SUSE offices in Nuernberg and we were able to successfully do a small presentation and demo to a group of interested SUSE people.

Amin:

I don't remember a distinct "greatest moment". But what is a really great feeling is to see that a project catches on, that other people get involved, use the code you have written and improve it in ways that you haven't thought of initially.

Tobias:

Hmm, also hacking on OpenSync/KitcheSync is much fun in general, the greatest moment was when the new KitchenSync frontend synced two directories via OpenSync the first time. But it was also cool when we managed to get the IrMC plugin working again after porting it to OpenSync.

> As we now know the worst problem you faced and your greatest moment, the only one missing is: What was your weirdest experience while working on OpenSync?

Cornelius:

Not directly related to OpenSync, but pretty weird was meeting a co-worker at the Amsterdam airport when returning from the last OpenSync meeting. I don't know how high the chance is to meet somebody you know on a big random airport not related at all to the places where you or the other person live, but it was quite surprising.

Tobias:

Since my favorite language is C++, I was always confused how people can use plain C for such a project, half the time your are busy with writing code for allocating/freeing memory areas. Nevertheless Armin did a great job and he is always a help for solving strange C problems :)

3. Devices and Programs

> Now I'd like to move on to some more specific questions about current and planned abilities of OpenSync. As first, I've got a personal one: I have an old iPod sitting around here. Can I or will I be able to use a program utilizing OpenSync to synchronize my calendars, contacts and music to it?

Cornelius:

I'm not aware of any iPod support for OpenSync up to now, but if it doesn't exist yet, why not write it? OpenSync makes this easy. This is a chance for everybody with the personal desire to sync one device or another to get involved.

Amin:

I don't think that there is iPod support yet for OpenSync. But it would definitely be possible to use OpenSync for this task. So if someone would like to implement an iPod plugin, I would be glad to help :)

> Which other devices do you already support?

Cornelius:

At this time, OpenSync supports Palms, SyncML and IrMC capable devices.

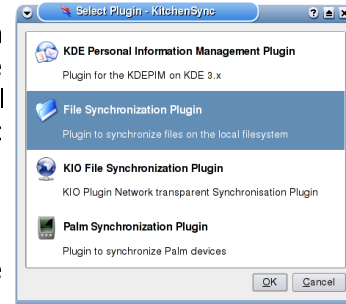
> Which programs already implement OpenSync and where can we check back to find new additions?

Cornelius:

On the application side there is support for Evolution [GNOME] and Kontact with KitchenSync [KDE] on the frontend side and the backend side and some more. I expect that further applications will adopt OpenSync once the 1.0 version is released.

Amin:

Besides kitchensync there already are a command line tool and a port of the multisync GUI. Aside from the GUIs, I would really like to see OpenSync being used in other applications as well. One possibility for example would be integrate OpenSync into Evolution to give users the possibility to synchronize their devices directly from this application. News can generally be found on the OpenSync web site (<http://www.opensync.org>).



Plugin selection in KitchenSync

4. Technical Dive

> So far for questions for users. I think it is time to give the developers something to devour, too. I'll keep this as a short twice-fold technical dive before coming to the takeoff question, even though I'm sure there's information for a double-volume book on technical subtleties.

As first dive: How did you integrate OpenSync in KitchenSync, viewed from the coding side?

Cornelius:

OpenSync provides a C interface. We wrapped this with a small C++ library and put KitchenSync on top. Due to the object oriented nature of the OpenSync interfaces this was quite easy.

Recently I also started to write a D-Bus frontend for OpenSync. This also is a nice way to integrate OpenSync which provides a wide variety of options regarding programming languages and system configurations.

> And for the second, deeper dive: Can you give us a quick outline of those inner workings of OpenSync, from the developers view, which make OpenSync especially viable for application in several different desktop environments?

Cornelius:

That's really a question for Amin. For those who are interested I would recommend to have a look at the OpenSync website. There is a nice white paper about the internal structure and functionality of OpenSync.

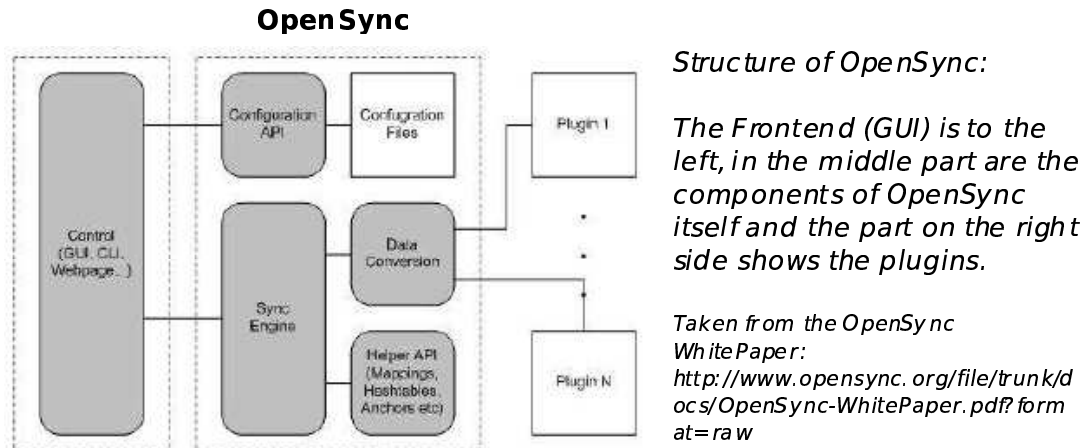
<http://www.opensync.org/file/trunk/docs/OpenSync-WhitePaper.pdf?format=raw>

Amin:

OpenSync consists of several parts:

First there is the plugin API which defines what functions a plugin has to implement so that OpenSync can dlopen() it. There are 2 types of plugins:

A sync plugin which can synchronize a certain device or application and which provides functions for the initialization, handling the connection to a device and reading and writing items. Then there is a format plugin which defines a format and how to convert, compare and detect it.



The next part is a set of helper functions which are provided to ease to programming of synchronization plugins. These helper functions include things like handling plugin config files, HashTables which can be used to detect changes in sets of items, functions to detect when a resync of devices is necessary etc.

The syncing logic itself resides in the sync engine, which is a separate part. The sync engine is responsible for deciding when to call the connect function of a plugin, when to read or write from it. The engine also takes care of invoking the format conversion functions so that each plugin gets the items in its required format.

If you want more information and details about the inner workings of OpenSync, you should really visit the [opensync.org](http://www.opensync.org) website or ask its developers.

5. Takeoff

> *To add some more spice for those of our readers, whose interest you just managed to spawn (or to skyrocket), please tell us where they can get more information on the OpenSync Framework, how they can best meet and help you and how they can help improving sync-support for KDE by helping OpenSync.*

Cornelius:

Again, the OpenSync web site is the right source for information. Regarding the KDE side, the kde-pim@kde.org mailing list is probably the right address. At the moment the most important help would be everything which gets the OpenSync 1.0 release done.

[And even though] I already said it, it can't be repeated too often: OpenSync will be the one unified syncing solution for the free desktop. Cross-device, cross-platform, cross-desktop.

It's the first time I feel well when thinking about syncing ;-).

Amin:

Regarding OpenSync, the best places to ask would be the opensync mailing lists at sourceforge or the #opensync irc channel on the freenode.net servers.

There are always a lot of things where we could need a helping hand and where we would be really glad to get some help. So everyone who is interested in OpenSync is welcome to join.

> *Many thanks for your time!*

Cornelius:

Thanks for doing the interview. It's always fun to talk about OpenSync, because it's really the right thing.

Amin:

Thank you for taking your time and doing this interview. I really appreciate your help!

Tobias:

Thanks for your work. Publication and marketing is something that is really missing in the open source community. We have nice software but nobody knows ;)

*Further Information on OpenSync can be found on the OpenSync Website:
<http://www.opensync.org>*

*This Interview was written by Arne Babenhauserheide
on behalf of himself, the OpenSource Community, SpreadKDE.org and the Dot (dot.kde.org).*

*It is licensed under the cc-attribution-sharelike license.
<http://creativecommons.org/licenses/by-sa/2.5/>*

I hope it got your mouth wet on the thought of the upcoming OpenSync.

*And if you want to make me happy, just take a look at my website and read some of my
shortstories, poems and strange ideas or listen to some of my songs.
Or donate me 10 M€, so I will never have to work again and can commit all my time to my hobbies ;)
- Arne Babenhauserheide (<http://draketo.de>)-*