

This is a **low resolution, black and white** version of the article you downloaded. To download the whole of Free Software Magazine in *high* resolution and color, please subscribe!

Subscriptions are free, and every subscriber receives our fantastic weekly newsletters — which are in fact fully edited articles about free software.

Please click here to subscribe:

<http://www.freesoftwaremagazine.com/subscribe>

# Interview with Simon Peter, creator of klik

Klik is a fantastic system which allows users to install an application without worrying about dependencies and without being root. I was lucky enough to talk to its author.

Tony Mobily

**T** M: Hello Simon! Please tell our readers about yourself...

Since this is a computer magazine, I should probably start with my computing history: I used one of the first Macs when they came out, later built my own PC at the expense of having to use Windows, switched to Linux in 2002. I'm 27 years old, native to Germany, and I studied Economics and Business Administration in Germany and the US.

**TM: Can you briefly explain what klik is?**

It's probably the easiest way to download and run software on your PC.

Simon Peter



**TM: When did you start working on klik? What inspired you?**

I had the idea in April 2003 when I was thinking of ways to use additional software that didn't come with the Knoppix Live CD. Back then, I had no idea of how it could be accomplished. In January 2004, klik came into existence, and since then it has been growing into what it is today, thanks to the feedback and help of many users. Today, there is a community of contributors and supporters on the #klik channel on irc.freenode.net, and together we help to drive klik forward.

-----  
Klik's probably the easiest way to  
download and run software on your PC  
-----

The ease of the first Mac inspired me: On the Mac in 1986, you could simply copy an application file to your hard disk and it would run. If you wanted it no more, you could simply move the application icon to the trash. In contrast, Linux 2003 required you to become root, type in cryptic commands, and understand what "broken packages" meant. And you never knew what went on "behind the scenes": what files were installed all over the place on the system.

I just wanted to have the same simplicity on Linux that the original Mac had. Plus, I wanted to store my applications wherever I had some space left—be it on a CD-ROM, a DVD, or a USB stick. Linux couldn't do that: it required me to install everything to /usr. I felt like the machine was in

command of me, not the other way around (as it should be). I felt it was time for a radically different approach: One file per application.

**TM: Software installation in Linux is renowned for being susceptible to the “dependency hell” problem. In your documentation, you write: “No dependencies besides base system. No other dependencies are necessary besides what comes already with the base system (distribution).”. However... sometimes, what “comes already with the base system” depends on what the user installed in the first place. I might not have libjpeg in my system, and klik application might need it. What happens in that case? Is there an automatic way of dealing with this sort of problem?**

Klik assumes that a certain set of libraries is part of your base system. Those are the libraries that are required by many applications and thus come with every major desktop distribution (e.g., libstdc++5, QT, KDE/GNOME, etc.). More “exotic” libraries that cannot be expected to be part of a user’s base system are downloaded by klik as part of the respective application.

**TM: Do applications tend to use more RAM when they run, as they are bundled with all the dependencies?**

No, they don’t, since klik uses shared libraries like any Linux distribution. It simply downloads the required libraries along with the applications, so that they are available in case they are not already part of the base system. By the way, applications aren’t bundled with all dependencies, but only with those that cannot be expected to be part of the base system. Is this a waste of hard disk space? No, since klik applications are stored in compressed form, they often use even less space than traditionally installed applications.

-----  
 On the Mac in 1986, you could simply  
 copy an application file to your hard disk  
 and it would run  
 -----

Really, most users don’t care about dependencies as long as their software works. My favorite story is about the 71 year-old grandmother who uses klik (<http://dot.kde.org/1126867980/1126871329/>)...

**TM: How is the base system made aware of the fact that a klik application is installed?**

One of the design principles of klik is that applications you download shouldn’t change the base system. Hence, klik doesn’t register file types, or change anything else on your system. This means that you need to open documents from within the respective application, or drag them onto the application’s icon. (This is a feature, not a bug—klik is designed not to ever “mess” with your system, regardless of how many applications you try out.)

**TM: You say you see this as a “feature”. However, I believe people will always want to be able to double-click on a document, for example, and start OpenOffice automatically (even though OpenOffice is installed as a klik application). I guess this is up to the distributions (in particular, Nautilus and Konqueror should be able to do that). Do you see this as an important point? Do you know if GNOME and KDE already allow for this, or if they will?**

You can set up GNOME/KDE to associate files with klik applications, just like with any other application. But that’s not the point: you then lose the ability to move your applications, e.g. onto a CD-ROM. Plus, if you have multiple versions of an application or multiple applications that work with a certain type of file, it would probably get more confusing rather than doing any good. Personally, I always open files with the open dialog of the respective application. If you don’t like that, you can drag documents to the respective application’s icon. Or, if you really want, you can set up a file association yourself.

Klik is intentionally designed in a way that doesn’t “register” applications with the base system. Only in this way, can you try out software and be assured that it doesn’t change your base system. I really dislike systems that “register” everything, because this tends to mess with your system, especially if you frequently try out lots of new software.

**If each application is contained in a mountable image, how do maintainers set what icons are associated with that application?**

This is neither necessary nor possible. (In the near future, KDE and Gnome will be able to display the icon that is contained inside the compressed image, just like they can display a thumbnail of a JPG file.)

**TM:** So, in other words, the CMG file will contain an image in a pre-determined spot, and Konqueror (KDE) and Nautilus (GNOME) will be able to extract it?

Yes. But “display”, not “extract”. The image will be displayed from its location within the .cmg file. (This way, the one app = one file approach still holds.)

**TM:** Every time I run an application, the system will mount the associated file system in /tmp/app. Does this mean that I will have ten mounted file systems if I run 10 applications? Could this cause problems if an application crashes (and the file system stays mounted!)? How does klik deal with these sort of issues?

Currently, klik mounts the compressed images in /tmp/app. As soon as the respective application closes (or crashes), klik unmounts the image. (So far the theory. In practice, when an application stays in an infinite loop, klik can’t unmount the image—then you have to kill the application manually. But this is true in any case.)

-----  
 One of the design principles of klik is  
 that applications you download shouldn’t  
 change the base system  
 -----

**TM:** In order for klik to work, you add seven entries in /etc/fstab. Each running application will “use” one entry. Does this mean that you can run seven applications at the most?

Correct. By default, the kernel is configured in a way that allows only for eight loop-mounts (klik leaves one as reserve). But this number can be easily increased using a kernel option. You’ll find the details in the klik wiki ([http://klik.atekon.de/wiki/index.php/Troubleshooting#When\\_I\\_try\\_to\\_run\\_more\\_than\\_7\\_klik\\_apps.2C\\_I\\_get\\_mounting\\_errors](http://klik.atekon.de/wiki/index.php/Troubleshooting#When_I_try_to_run_more_than_7_klik_apps.2C_I_get_mounting_errors)).

**TM:** Is there an alternative to adding entries to /mnt/fstab, or an alternative to mounting altogether?

Yes, it’s called userland-mounting (FUSE), and will be used in future distributions that support klik, such as the next release of Kanotix. This will remove the need for fstab entries altogether.

**TM:** How does klik deal with upgrading?

Klik has no automatic mechanism for upgrading. When you feel that you’d like a newer version of an application, you just klik it again. (You can keep both the old and the new version at the same time—which makes klik pretty unique!)

**TM:** So, you type `klik://xviera`, you end up downloading the program Commute. This seems to confuse some people. What happens internally when you type `klik://xviera`? And... why is it a host name cannot be specified in such a URL? Does this mean that everything that comes from `klik://` will come from klik’s repository? How does that work if a program is available for several distributions?

To browse all existing applications, please go to our home page (<http://klik.atekon.de/>)—there you will find a large “warehouse” with thousands of applications, most of them with descriptions, user comments, and even screenshots.

`klik://` URLs contain no host name. This is an important security feature because it prevents klik from downloading software from random places. Klik only downloads so-called “recipes” from the klik server. Recipes tell the klik client what software to download (and from where), and how to re-package it as a .cmg file. Most of the time, one recipe fits all supported distributions (so no need for different packages for different distributions).

**TM:** Companies wanting to distribute proprietary software have had a problem with Linux, because there hasn’t been an easy and universal way of distributing an application. They could choose between tar.gz, tar.gz containing proprietary installers, RPM (for several distributions), debian packages... But now, they can just give away a .cmg file that will “just work”. Is this correct? Do you think companies will start doing that soon? (I am thinking for example of interactive CDROMs, CD dictionaries, etc.).

The easiest thing for software publishers to do is put up either a .deb, .rpm, or .tgz package. Klik maintainers can then write a klik recipe (basically just entering the download URL and some easy instructions). This is the way in which `klik://skype`, `klik://realplayer10`, and `klik://opera` work. Note that the software is still downloaded from the original vendor’s web site, klik does not need to store or

host a .cmg file since it is created by the klik client on-the-fly.

**TM: So... you don't think vendors would benefit by distributing CMG files rather than the "usual" ones?**

Yes, sure. But instead of pushing vendors to learn and adopt the .cmg format themselves, they can just continue to do what they do already, and klik does the rest—I'm not sure that we'd have Skype klik-able today if we needed active vendor involvement for it.

**TM: What about vendors who give away interactive CDs and would like to have them available for Windows, OS X and Linux for example? Wouldn't a .cmg file be the solution? (Besides, there could be no recipe for an application that is distributed in CD format!)**

Sure, that could be interesting. (The .cmg format isn't finalized yet though, and is still subject to change.)

**TM: I see klik as a major revolution for GNU/Linux. With klik, everything is changed: people can now install whatever they like with no effort and without root access, AND companies can distribute proprietary software that works on multiple distributions with little to no effort. The response you are having seems to confirm my feeling: you did indeed start a revolution. Well... how does that feel?**

Sounds like a plan! ;-)

## Bibliography

[1] klik's home page (<http://klik.atekon.de/>)

## Copyright information

© 2006 Tony Mobily

Verbatim copying and distribution of this entire article is permitted in any medium without royalty provided this notice is preserved.

### About the author

Tony Mobily is the Editor In Chief of Free Software Magazine (<http://www.freesoftwaremagazine.com>).