

08/09/09 - GSoC, week 11: building instructions

Disclaimer: the following guide may overwrite some of the files installed through the standard packages of your distribution, so it may be better if the work is done on a newly installed machine different from your production one, or in a different directory along with kdesupport, kdelibs and kdebase from SVN trunk. You have been warned...

As promised, while I'm currently tweaking a few things out in the dialogs, here there are the compilation instructions for trying my code. No .deb precompiled packages (yet?), a couple of passages were a little more complicated in relation to the time that I can give them, so you need to get a little dirty, but I'm sure that this is not a problem.

As a premise, I assume you are a little confident with compiling sources, and you know what cmake is; given that, here we go!

The following passages work for sure in Ubuntu Jaunty, updated to KDE 4.3 (oh, by the way: you need KDE 4.3 or better (that is: trunk) for all this to work), and I know this because my new shining laptop arrived last wednesday, so I did this to work on my code.

Prerequisites

You need at least kdelibs and kdebase packages (let me repeat: minimum version is 4.3, so if you are on Jaunty, you do need the backports repository and the upgrade operation, please refer to www.kubuntu.org for instructions), plus their development versions: kdelibs5-dev and kdebase-workspace-dev (pay attention: kdebase-dev is from 4.2.2); for any other missing library, cmake will tell you what you need.

Source code

Premise: if you don't have the latest version of Strigi, you need to get it from SVN, too: for example, in Jaunty is 0.6.4 and in Karmic, at least a couple of days ago, was 0.6.5, but you need at least 0.6.95. So, if you need it, then get it:

```
svn co svn://anonsvn.kde.org/home/kde/trunk/kdesupport/strigi
cd strigi
mkdir build
cd build
cmake ..
```

```
make  
(as root) make install
```

You can install it in /usr/local, but this thing is probably going to stop your strigi filesystem analysis: if you don't care about it, then go and compile it; otherwise, I think you will need to install trunk in some different directory and work with it.

Then, you need to checkout from SVN [Adam's kioslave](#):

```
svn co svn://anonsvn.kde.org/home/kde/branches/work/soc-virtualfolders  
cd soc-virtualfolders  
mkdir build  
cd build  
cmake -DCMAKE_INSTALL_PREFIX=/usr ..  
make  
(as root) make install
```

As you may notice, you need to install that code in the same folder you have KDE, unfortunately it does not seem to work in /usr/local or similar folders (but any suggestion in this sense is accepted): as a consequence, it will overwrite the current kioslave from your packages. Again, if you don't like it, then compile trunk in a directory different from the one of your standard installation.

After the last command, everything should work.

Now, let's go with playground:

```
svn co svn://anonsvn.kde.org/home/kde/trunk/playground/base/nepomuk-kde  
cd nepomuk-kde  
mkdir build  
cd build  
cmake ..
```

Ok, now you have your building makefiles ready; usually I only compile what I really need from this code, also because there is the need to install the annotation plugins, because they are searched at runtime by the dialog system, so here it is the sequence of commands to compile and install them and their dependencies:

```
cd kcompleter  
make  
make install
```

```
cd ../pimo
make
make install
cd ../nepomukcompletion
make
make install
cd ../scribo
make
make install
cd ../annotationplugins
make
make install
cd ..
```

Ok, with this you should have installed everything you need (of course, if needed, install commands have to be launched as root); now we can build smartsave and test it:

```
cd smartsave
make
make install
```

The installation passage is optional, because it will not install the testing application.

If you want to try the dialog, enter the test directory and launch the texteditor app: it is just the same simple notepad from Techbase tutorials, but with my open and save dialogs instead of the standard KDE ones.

Troubleshooting

Some things can go wrong: the first one is that you see the standard dialogs in the test app, and if this happens, then you probably have shut down Nepomuk, please restart it from the system settings dialog.

The second one is that you don't get any suggestion in the save suggestion box: if this is true, then you can try by restarting the kde daemon, by invoking *kdeinit4* from a console, so that the plugins installed before are registered.

The third one is that you cannot see any file in the open dialog: if this happens, then you have to check that the kioslave has been correctly installed in your KDE main directory (/usr on a standard system, or something else if you work directly on

trunk).

What now?

Now that you have a text editor with the dialogs, what can you do? Well, I'm going to prepare a screencast, probably in the following week, to show a couple of features; bear in mind that there are some tweaks on which I'm working on, so this is not a perfect system or a system ready to be shipped in a stable release (far from that), so be patient and test, and feel free to report any kind of trouble you may find along the way.

GSoC status

Well, one week left: the dialog system surely has a base on which to work for both UI and some backend code; there are a couple of things that I'd like to work out before August the 17th; then I may try to add these dialogs to a real KDE application, instead of a simple test program, but we will see what will come up in the next few days...