Subject: U++ 2008.1 rpm automatic reconstruction for the rpm maintainer
Posted by amrein on Fri, 15 Aug 2008 22:20:06 GMT
View Forum Message <> Reply to Message

(UPDATED BECAUSE AT LEAST THERE ARE 2 PATCHES NEEDED TO PACK UPP 2008.1 CORRECTLY)

Note : There are binary rpm and easier instructions for rpm build here:
http://www.ultimatepp.org/forum/index.php?t=msg&th=3734& amp;start=0&

Hi,

We talked about U++ rpm release. Here is a rpm .spec file. I built U++ on Mandrivalinux (www.mandriva.com).

Here how to use and test it (you run Ubuntu I guess). Install steps:

. Install the kernel devel source + virtualbox in Ubuntu (apt-get install virtualbox I guess).
. Get the Mandriva 2008.1 install CD from their website.
. Start virtualbox and set it to boot on the downloaded CD.
. Install Mandrivalinux.
. As root in Mandrivalinux, install rpm-build (sudo won't work out of the box here for security reason):

```
# su
# urpmi rpm-build
# urpmi freetype2-devel gtk2-devel pango-devel atk-devel cairo-devel X11-devel xft2-devel expat-devel
# exit
```

Now, we need to prepare the rpm build environment. You need to do this only once:

```
# cat > ~/.rpmrc << EOF

buildarchtranslate: i386: i586
buildarchtranslate: i486: i586
buildarchtranslate: i586: i586
buildarchtranslate: i686: i586
EOF

# cat > ~/.rpmmacros << EOF
%_topdir              $HOME/rpm
%_tmppath             $HOME/rpm/tmp

%_signature           gpg
%_gpg_name            Mandrivalinux
```

```
%_gpg_path            $HOME/.gnupg
%distribution         Mandrivalinux
%vendor               Ultimate++ team
%packager             YourName YourLastName <youremail@provider.location>
EOF
```

```
# mkdir -p     ~/rpm/{BUILD,RPMS/{i586,x86_64,noarch},SOURCES,SRPMS,SPECS,t mp}
```

Now, we download what we need:

- First the spec file

```
# scp your-pc-ip:~/where-you-downloaded-the-spec-file/upp.spec ~/rpm/SPEC/
```

- The U++ tarball:

```
# scp          your-pc-ip:~/where-you-downloaded-upp-src-2008.1.tar.gz/upp- src-2008.1.tar.gz
~/rpm/SOURCE/
```

- The .desktop patch (because there's a bug in the upp2008.1 package). See next posts to find it.

```
# scp          your-pc-ip:~/where-you-downloaded-the-patch/upp-src-2008.1.f
ix_png_name_in_desktop_file ~/rpm/SOURCE/
```

- The correct theide-48.png icon (because the provided one in the tarball is corrupted). See next posts to find it.

```
# scp  your-pc-ip:~/where-you-downloaded-the-icone/theide-48.png ~/rpm/SOURCE/
```

So, now, you have rpmbuild ready to do its job. You have the build script (spec file). You have the tarball a patch and a png.. Let's build the rpms and install it:

```
# rpmbuild -ba ~/rpm/SPEC/upp.spec
```

To install your new rpm package:

```
# urpmi ~/rpm/RPMS/i586/upp-2008.1-1.i586.rpm
or
# urpmi ~/rpm/RPMS/x86_64/upp-2008.1-1.x86_64.rpm
```

You can than run theide from the command line or find it into Menu/Development/

Now you know out to have a rpm build system. You can use cron job to have a daily building system running. You know how to start a script at boot time to automate the build of new daily release and then send result of the build and errors and then shutdown the virtual machine (just shutdown -h now) to let the server use its processors for something else.

If not, well... do you like man page and google?
I'm joking.

Have a look in the spec file. You will understand why I said there are a few disturbing things into the U++ tarball.
If you want more information about rpm build, have a look here:

http://wiki.mandriva.com/en/Development/Howto/RPM
http://wiki.mandriva.com/en/Development/Packaging

Other issue: When you run theide, their is no template available without GCC.bm. The spec file %file section doesn't have more than one icon. theide-48.png is not a valide png. The theide-48.png I added comes from the debian script (they create it manually). The desktop file doesn't need to have theide.png in the "icon=". just theide without the extention. The patch I provide fix it.

Type this to know the standard Makefiles names for paths, CFLAGS, CXXFLAGS in linux:

rpm --eval %configure

and

rpm --eval %makeinstall

You can now also change your Makefile accordingly.

Regards and happy coding
Amrein-Marie Christophe

File Attachments

1) upp.spec, downloaded 696 times

Subject: Re: U++ 2008.1 rpm
Posted by mirek on Sat, 16 Aug 2008 07:32:39 GMT
View Forum Message <> Reply to Message

Good...

Are you willing to extend the team and become rpm maintainer?

Well, as you might have guessed, the final near-term goal is to setup nightly automated builds.

Which makes me wonder how well nightly builds will work together with virtualbox. Any idea?
(IMO, the problem is that we need to invoke script in virtual machine).

Would it be eventually possible to do this using chroot instead?

Mirek

Subject: Re: U++ 2008.1 rpm
Posted by amrein on Sat, 16 Aug 2008 13:11:21 GMT
View Forum Message <> Reply to Message

(PATCH IS AT THE END OF THIS MESSAGE)

luzr wrote on Sat, 16 August 2008 09:32Good...

Are you willing to extend the team and become rpm maintainer?

Why not. But I just have a few issues with U++ and I would like to be sure they will be addressed before giving long contribution to the team (more time than already given to show my interest to this project). I'm talking about the web site, the licence, the doc, the source directory structure, the "static link only" with no official dynamic library and finally the Linux port status.

Quote:Well, as you might have guessed, the final near-term goal is to setup nightly automated builds.

Which makes me wonder how well nightly builds will work together with virtualbox. Any idea? (IMO, the problem is that we need to invoke script in virtual machine).

When the virtual machine is installed then shutdown, the virtual disk is still there and keep all modifications.
"VBoxManage" is a command line tool. You can manage the Virtual box with it or with the graphical front end "VirtualBox".

If you call one of your VM "Mandriva Linux", you can start it like this:

VBoxManage startvm "Mandriva Linux"

VBoxManage will load and run the VM.

You can share files using NFS, samba, ssh, ...

To be short, with a VM, it's like having another PC running another OS. Running "VBoxManage", it's like switching it on.
So, you just need to do on this "new PC" is what you would like it do do. Something like creating the rpm build environment first than get the daily tarball or create the daily tarball after a "svn up" command, then rebuild the rpm. There are faster solution of course but more complicated to keep the distro up to date.

(urpmi doesn't won't work here in a chrooted environment since 2008.0 for an unknow reason).

Quote:
Would it be eventually possible to do this using chroot instead?

Mirek

Yes of course. You know about chroot? Than you can do something a bit faster. Most rpm distributor use it.

All distributions doesn't use the same libraries version nor the same compiler version. You can't just install debian packages + rpm to deal with the rpm. So you will have to install a rpm based distro (on a VM or another partition of your hard drive), then copy this new root partition and U++ source into a local directory. Before chrooting into it, "mount proc" then "chroot ." then compile U++. Don't forget output redirection of stdout and stderr to know if something had going wrong, ...

File Attachments

1) upp-src-2008.1.fix_png_name_in_desktop_file, downloaded 562 times

---

Subject: Re: U++ 2008.1 rpm
Posted by mirek on Sat, 16 Aug 2008 13:31:40 GMT
View Forum Message <> Reply to Message

amrein wrote on Sat, 16 August 2008 09:11
I'm talking about the web site

There is always a room for improvement. The problem is lack of manpower to do them.

Quote:
, the licence

GPL or LGPL would ruin U++ business-wise.

MIT is the most likely.

Quote:
, the doc

Working on it. No doxygen for several reason (reconsidered many times).

Quote:
, the source directory structure,

Do not expect this to change. It is one of main selling points of platform.

Besides, do not expect U++ to "play by rules" in all cases. That is the point.

Quote:
 the "static link only" and but no official dynamic library,

"static link only" is only the most logical and straightforward path.

I would like somebody started "official dynamic library". The only problem is that:

- I am not sure HOW to do that. U++ is highly modular, it will be difficult to decide how these modules translate to .so (having 50 libraries out of U++ does not seem feasible

- Personally, I am not even INTERESTED in doing it. It does not solve a single problem and only adds a couple of them, at least for me.

(the fact that I am not interested does not mean I would not support such thing).

Thanks about info about virtualbox etc...

I have now to consider the next "infra" phase - those nightly builds.

In fact, another problem there is how to make possible for maintainers to maintain releases via svn. I suspect we should use something like specific folder in svn, which gets exported each night and "run-parts" applied on it...

Mirek

Subject: Re: U++ 2008.1 rpm
Posted by amrein on Sat, 16 Aug 2008 22:08:50 GMT
View Forum Message <> Reply to Message

(THE CORRECT THEIDE-48.PNG IS AT THE END OF THIS MESSAGE)

luzr wrote on Sat, 16 August 2008 15:31amrein wrote on Sat, 16 August 2008 09:11
I'm talking about the web site

There is always a room for improvement. The problem is lack of manpower to do them.

Quote:
, the licence

GPL or LGPL would ruin U++ business-wise.

MIT is the most likely.

http://www.opensource.org/licenses/mit-license.php
http://www.opensource.org/licenses/bsd-license.php

I asked a question (for free... hard to achieve) to a lawyer of my ex employer (a friend). His short answer is:

BSD license = Yours + "If you modify U++ source, neither the name of the U++ main teams and contributors may be used to endorse or promote products derived from this software without specific prior written permission."

BSD license = Do whatever you want but: (1) if you release modified source code, don't use our names (2) if you release the binary generated with the unmodified source code and your software doesn't link with it, you must provide the original and complete source code of our software too.

MIT license = Yours + "The above copyright and license must be included in all copies or substantial portions of any released source code." - "If the binary is provided as this, if your software doesn't link with it, if you have made no modification into U++ source code, you must publicly acknowledge the use of the software in your software/doc and must make the U++ source code available publicly."

MIT licence = Do whatever you want but if you release a file including part of our source code, this license and the copyright holders (U++ team and contributors) must be in the source file including our source code.

Quote:
Quote:
, the doc

Working on it. No doxygen for several reason (reconsidered many times).

It would be so cool to have topic++ been able to help people edit doxygen documentation directly into the source. A complete doxygen editor + topic++ (replacing doxygen completely).

It's hard to maintain source code documentation. It's also hard to come back to old source and not find any // or /* */ to comment the source. As soon as you are more than 2 developers, adding comments in source code is really welcome.

Quote:

Quote:
, the source directory structure,

Do not expect this to change. It is one of main selling points of platform.

Besides, do not expect U++ to "play by rules" in all cases. That is the point.

Sure, or it wouldn't be able to evolve quickly.

Quote:
Quote:
 the "static link only" and but no official dynamic library,

"static link only" is only the most logical and straightforward path.

I would like somebody started "official dynamic library". The only problem is that:

- I am not sure HOW to do that. U++ is highly modular, it will be difficult to decide how these modules translate to .so (having 50 libraries out of U++ does not seem feasible

- Personally, I am not even INTERESTED in doing it. It does not solve a single problem and only adds a couple of them, at least for me.

(the fact that I am not interested does not mean I would not support such thing).

Thanks about info about virtualbox etc...

I have now to consider the next "infra" phase - those nightly builds.

In fact, another problem there is how to make possible for maintainers to maintain releases via svn. I suspect we should use something like specific folder in svn, which gets exported each night and "run-parts" applied on it...

Mirek

You welcome.

Many thanks for all those answers.

Concerning svn, you can use user groups to prevent someone from touching part of the repository. You could create different branches (like Linux kernel, like upp2008.1_amrein) and give them full access only to those branches (using group policy).

The big problem: this is not open enough to be interesting and also hard to maintain. This project

is not big enough for multiple branches. Having separated branches will cause problems for merging if contributors go in too different paths. This is less man power in the main trunk. Anyway, if really required, for new feature for example, it can be interesting to have another branch, but without big permission restriction.

Anyone can play at home with a svn local copy for small patch. Here is how most FOSS teams work:

- Other FOSS project give permission in a meritocracy way.
- Only frequent and trusted contributors have write access.
- External contributors (anyone) can send svn diff (better for trunk) or just diff. They send them via the mailing list, the bug tracker or via email if any.
- If a contributor with write access want to do something bigger than a small patch (move/rename files, change the way TheIDE works, ...), he must discuss it first publicly and get approval from the team + the svn/project admin.
- They use groups to prevent modifications only if they have very critical parts and they give write access to those parts only to senior contributors.

File Attachments

1) theide-48.png, downloaded 864 times



Subject: Re: U++ 2008.1 rpm
Posted by mirek on Sat, 16 Aug 2008 23:41:26 GMT
View Forum Message <> Reply to Message

amrein wrote on Sat, 16 August 2008 18:08
- Other FOSS project give permission in a meritocracy way.
- Only frequent and trusted contributors have write access.
- External contributors (anyone) can send svn diff (better for trunk) or just diff. They send them via the mailing list, the bug tracker or via email if any.
- If a contributor with write access want to do something bigger than a small patch (move/rename files, change the way TheIDE works, ...), he must discuss it first publicly and get approval from the team + the svn/project admin.
- They use groups to prevent modifications only if they have very critical parts and they give write access to those parts only to senior contributors.

Well, this is pretty much how it worked here and will work. The only problem is that there were only 3 trusted contributors and we have outgrown it....

BTW, this is also one of reasons why doxygen comments are not viable option... (but there is

something possibly even better coming).

Mirek

Subject: Re: U++ 2008.1 rpm
Posted by mirek on Sat, 16 Aug 2008 23:45:04 GMT
View Forum Message <> Reply to Message

amrein wrote on Sat, 16 August 2008 18:08
Concerning svn, you can use user groups to prevent someone from touching part of the repository. You could create different branches (like Linux kernel, like upp2008.1_amrein) and give them full access only to those branches (using group policy).

Well, in fact, we will be using this, but not for branches, but for trunk parts. And in reality, it is not very complicated.

BTW, we have "stable trunk" policy here. trunk is always considered stable enough to be used in mission critical apps. It should always be more stable than latest "official" release.

Mirek

Subject: Re: U++ 2008.1 rpm
Posted by amrein on Sun, 17 Aug 2008 01:23:51 GMT
View Forum Message <> Reply to Message

luzr wrote on Sun, 17 August 2008 01:41
Well, this is pretty much how it worked here and will work. The only problem is that there were only 3 trusted contributors and we have outgrown it....

Doc + website + MPL or the officially modified BSD license + dynamic linking (for FOSS dev) + static linking (for FOSS and proprietary dev) and the project will grown further.

You want me as the rpm builder. Let's try it if you are still interested.

As soon as I get a few things working correctly with TheIDE + a dynamic library, it will be easy to include TheIDE + U++ in Mandrivalinux official contrib repository to get it main stream.

Quote:
BTW, this is also one of reasons why doxygen comments are not viable option... (but there is something possibly even better coming).

Mirek

ArGhhhh... and you don't tell us more? :'(

It's a shame !!!

Subject: Re: U++ 2008.1 rpm
Posted by amrein on Sun, 17 Aug 2008 02:57:04 GMT
View Forum Message <> Reply to Message

I need to generate GCC.bm for the rpm package as there is no .bm included in the tarball. Which configuration do want you me to use?

I'm asking this because ChkSupp() from uppsrc/ide/Install.cpp (line 254) search for GCC.bm and TheIDE won't work correctly without it.

File Attachments
1) upp-src-2008.1.fix_gcc_optimisation_flag_bug, downloaded 579 times

Subject: Re: U++ 2008.1 rpm construction [UPDATED]
Posted by amrein on Sun, 17 Aug 2008 19:47:09 GMT
View Forum Message <> Reply to Message

Ok. That doesn't matter. I updated the fist second and third post to include patches and newer spec file to have a working 2008.1 U++ package.