Subject: Maintaining release?
Posted by [mirek](#) on Sun, 27 Jul 2008 07:07:06 GMT

View Forum Message <> Reply to Message

Well, it seems like we are finally approaching the final release.

I think we could consider whether, for the first time, we should be 'maintaining' it, I mean fixing bugs and release 2008.1.1 .2 etc?

What would be the best model to do this?

Mirek

Subject: Re: Maintaining release?
Posted by [cbpporter](#) on Sun, 27 Jul 2008 10:19:30 GMT

View Forum Message <> Reply to Message

This is a good idea. I believe each dev version should be followed by an update to the core release after it has proven stable. This way we won't have so much trouble with people being afraid to download dev version's and maybe miss out on bug fixes and new features.

But why 2008.1.1? Why not 2008.2? Is there a reason for only a release per year? Or do you want to package a lot of new features in each release?

Subject: Re: Maintaining release?
Posted by [mirek](#) on Sun, 27 Jul 2008 14:17:00 GMT

View Forum Message <> Reply to Message

cbpporter wrote on Sun, 27 July 2008 06:19
Is there a reason for only a release per year?

Well, I guess the reason is that this was originally supposed to be 2007.2 release

And yes, after this is finally released, I plan to poke in too many things in TheIDE. I am afraid it will take quite a lot of time before next release...

Mirek

Subject: Re: Maintaining release?
Posted by [mr_ped](#) on Tue, 29 Jul 2008 08:10:17 GMT

View Forum Message <> Reply to Message

I think there *will* be need for "fixed" release after 2008.1, but I don't care whether you will call it 2008.2 or 2008.1.1 (I think .2 is easier, and the new features can be planned for .3 or 2009-2012

whatever)

So let's release 2008.1, make a branch on SVN, make sure it contains everything it is needed to build all platforms releases from that branch, and let's do any FIXING on that branch, after some time re-release it as 2008.2. (and eventually .3 or more)

In the meantime let the havoc happen on trunk to prepare next *new* release, and release it when it will be ready.

This way there's still one big problem, the development will be split between new version, and fixed 2008.1 one. With so few core developers this doesn't sound good, and I think the only way to not waste time is to do on the fixing branch just the really important system fixes, and guard everyone to not add anything to it. The development should proceed only at trunk branch, which can be than used to produce dev releases, which can be used by people who want to be on edge of development.
(although this will lead to situation where "fixed" 2008.2 released *after* some dev release will lack new features included in the dev release, so I think the naming of dev releases should reflect this in some way, if you can estimate the work needed for next new release and it looks reasonable to happen after 2008, the "dev2009.x" may save lot of confusion in such case)

That's my opinion.

Subject: Re: Maintaining release?
Posted by mirek on Tue, 29 Jul 2008 08:57:12 GMT
View Forum Message <> Reply to Message

mr_ped wrote on Tue, 29 July 2008 04:10I think there *will* be need for "fixed" release after 2008.1, but I don't care whether you will call it 2008.2 or 2008.1.1 (I think .2 is easier, and the new features can be planned for .3 or 2009-2012 whatever)

So let's release 2008.1, make a branch on SVN, make sure it contains everything it is needed to build all platforms releases from that branch, and let's do any FIXING on that branch, after some time re-release it as 2008.2. (and eventually .3 or more)

Well, I would not like to lose the unlikely chance to release more than one U++ major version in a single year

Mirek

Subject: Re: Maintaining release?
Posted by mr_ped on Tue, 29 Jul 2008 09:05:28 GMT
View Forum Message <> Reply to Message

I think you are way too much inside this project Mirek. For outsider it doesn't matter if 2008.2 is

just "fixed" 2008.1, and 2008.3 is true new release, for outsider the "2008.x" is the latest stable build which is recommended for usage.

If the new release has new features, it's exciting, but you have to do some additional testing if you may upgrade UPP without breaking your current projects. If it is "just" more stable, it's exciting, because you can safely upgrade, only (old) bugs in application can appear with such release.

So either way, every stable build is exciting and welcome.

edit: maybe you were ranting about dev2009 naming. Well, the dev releases had always weird names, you can return to that, call it dev80x so nobody can decipher (without reading roadmap and/or forums) how dev releases relate to that stable 2008.x.