

---

Subject: Re: The plan....

Posted by [amrein](#) on Fri, 29 Aug 2008 20:06:14 GMT

[View Forum Message](#) <> [Reply to Message](#)

---

luzr wrote on Fri, 29 August 2008 20:12amrein wrote on Wed, 27 August 2008 18:08

So 7 assemblies:

"nests", right?

Yes.

For me, those two words "Assembly" and "Nest" are complicated and should disappear. "Assembly", "Nest" and "Package" organisation could be break into "Collection", "Directory" and "Package" (when someone say "Garbage collector" in java, everyone understand what he means). I don't hate birds ("Nest"... ) but a few newcomers are still confused by those names (me included, I had to read the doc 4 times).

Example: in the first TheIDE window (when you open it), there are two rows "Assembly" and "Packages". That create confusion.

In fact, with this window, you are watching packages from the Nests of the default TheIDE Assembly right?

(If I'm wrong about all this, I hope I won't find a gun too close to me)

If you want to create another Assembly, you need at present to do it manually.

Now, lets break this into "Collection", "Directories" and "Packages":

- The main title of the window become "Select main package from "Default Collection" "
- First row title: "Directories"
- Second row title: "Packages"

This doesn't break the "Assembly" original idea. Anyone could create is own Collection/Assembly if he needs to.

Quote:Quote:

- myapps
- examples
- templates
- upp
- theide
- 3rdparty
- plugins

- What is the point of joining all 3 kinds of examples into single dir?

To clean directory listing in c:/upp. Could become for example:

- upp/myapps (your own packages)
- upp/examples (training packages)
- upp/manual (documentation for learning)
- upp/source (the core: libupp, tools and plugins)

With examples: "applications" = complete apps, "references" = small apps to demonstrate one feature, "manual" = examples from the manual.

With 3 kinds of examples in one directory, there is a need perhaps to add something else: be able to group subdirs. The main window in TheIDE use symbol like "(+)" to close or open group of files. Here, the idea is for all Nests without packages but having subdirs with packages to be able to open or close groups to see those sub-packages with "(+)"

Quote:

- What is "templates" good for?

It's where people can create or fix templates for TheIDE. It's the default TheIDE template directory. Could be in TheIDE directory though.

When you create your new application, TheIDE already asks you to choose one of them at present.

Another reason is to transform TheIDE into TheLinuxWindowsIDE with templates and support for many other toolkits. A tool you will want to use for any projects from DLL to .so, from Qt applications to GTK+, MFC, wxWindows.

I don't remember who said something like this: "Keep your friend close to you. Keep your enemies even closer.". They are not enemies, I know. The idea is to open TheIDE doors to all kind of developers. Become the most used IDE. If someone wants to create a new application, he should think TheIDE first.

Quote:

- Having separate nests for 3rdparty and plugins would lead to more complicated assemblies - you would have to add 3 nests instead of one. IMO it is better to have it as subdirs in the nest, which is current status.

- When they are from U++ team, sub directories in a packages are better.
- "3rdparty" is a directory containing packages not from U++ team. They could be useful for other applications.
- "Plugins" could be external sql plugins, image format plugins, codecs plugins. The idea of plugins is to open U++ to third party plugins and be able to handle them dynamically at runtime. They could link uplib to "3rdparty" too for example.

Last idea, with 3rdparty and plugins, is to be able to link TheIDE dynamically with 3rdparty lib not

in source/3rdparty but in /usr/lib. Why? Because 3rdparty applications receive security fix from main Linux distributor. This is a bit far from Windows security. No need to rebuild your application if the linux distributor has patched one of the dynamically linked library. No need to wait from next U++ release to get those packages fix.

Quote:

OTOH, to truly separate 3rdparty stuff, we MIGHT consider putting 3rdparty folder in uppsrc and keep only U++ interfacing stuff (-> our core) in "plugin". The grand question is whether is it worth the trouble...

Mirek

The main idea is be able to say: U++ is a "Collection" or "Directories" containing "Packages". TheIDE is one of those directories. IDE and Topic++ are packages from TheIDE directory.

More complicated:

A directory contain a complete project with many packages.

Subdirectories are named packages when they contain something to build with a .upp file.

Subsubdirectories are subpackages of packages.

---