Subject: Re: User lists of "bad" naming of classes, functions etc in U++...
Posted by cbpporter on Sat, 30 Aug 2008 12:05:48 GMT
View Forum Message <> Reply to Message

Quote:My problem: I compare "Assemblies" with Linux file system layout. "Nest" with LD_LIBRARY_PATH + PATH. "Packages" with SRPMS and their build dependencies.
To understand this concept I think you need to think a little out-of-the-box and forget the stupid concept of a module that has been created by using text inclusion in C/C++. You need to consider things from the standpoint of a modern and true module system.

But it could be that different persons have different understanding of what packages, nest and assemblies mean. Here is my take: packages define a group of classes and functions which are tightly related logically and which try to offer the means to solve a problem. Like Core offering a basic library of IO and data structures, CtrlCore offering a mechanism to create widgets and CtrLib offering a set of concrete widgets. Depending on definition, packages could also be considered modules, since U++ packages tend to be more general (i.e. Core being a basic module, which could contain packages like: IO, Containers, i18n, MemoryAllocator, MT,... (not the case though)). Packages are also independent, as in they don't require to physically contain other packages, and dependencies are simply referred to. Packages are also movable in the directory structure.

Nest are group of packages and can be considered modules or libraries. Depending on your need, they can be considered projects also, but more often I think they are group of projects (from a Visual Studio perspective).

Assembles are sets of nests which are used to build an application. For example, you could have the U++ nest, and external nest for ODF parsing and a nest that defines a GUI. You combine them in an assembly and are limited to these three nests together with their packages. By combining them, you get the source code equivalent of you application.

It would be really fun if by these statements I manage to shoot myself in the foot and have a completely wrong opinion about this layout . If that's the case, then Mirek, please don't give up and try to explain things so we can both get it.

But I agree that it is confusing. Especially the package chooser, where you get to choose your assembly, but have no idea what nests it includes. Also, there are assemblies formed out of one nest, and because nests have no name or visible entity, you can get confused. Also, by grouping all packages from all nests in you assembly alphabetically, the tight logical order which may be present in a nest is broken. Why do I have an IO package next to a GUI package? It would be nice if if nest where nameable and we could get a tree in the package chooser where nests are isolated on unique top level branches and packages are leaves.