Subject: [PROPOSAL] Rebuild only selected packages.
Posted by rxantos on Tue, 21 Jul 2015 17:41:18 GMT
View Forum Message <> Reply to Message

Sometimes when using blitz I have to rebuild the whole project I am working on. Otherwise I get errors when running a program.

No big deal, except that rebuild all rebuild the entire project. Right now I need to go one by one and clean my packages and then rebuild. Otherwise I get to wait for all the U++ libraries to be remade (which gets old pretty fast).

The ideal would be:
- Select all projects one wants to rebuild.
- Save that selection.
- Run a rebuild my files.

Is there a way to do this without editing the ide?
I tried using a Esc macro. But while there is a BuildProject function there is no CleanProject function nor is there a RebuildProject function.

Better yet. Have the option to not rebuild anything in the uppsrc directory. (aka do not clean uppsrc, only my workspace).

Did some of it:
In uppsrc/ide/Build.cpp
Add

```
void Ide::PackageRebuid()
{
 const Workspace& wspc = IdeWorkspace();
 int pi = GetPackageIndex();
 if(pi >= 0 && pi < wspc.GetCount()) {
  console.Clear();
  CleanPackage(wspc, pi);

  BeginBuilding(true, false);
  Vector<String> linkfile, immfile;
  String linkopt;
  bool ok = BuildPackage(wspc, pi, 0, 1, mainconfigparam, Null, linkfile, immfile, linkopt);
  EndBuilding(ok);
 }
}
```

in uppsrc/ide/Ide.h
Add

```
void  PackageRebuid();
```

In uppsrc/ide/ide.key
Add

KEY(REBUILDPACKAGE, "Rebuild package", 0)

In uppsrc/ide/idebar
Add

menu.Add(b, AK_REBUILDPACKAGE, THISBACK(PackageRebuid)).Help("Rebuild current package");

to the

void Ide::BuildPackageMenu(Bar& menu)

function.

This adds the ability to rebuild a package in one step instead of two.
I'm still working on the select what to rebuild part. Ideally it would be a:
- Clean package assembly
- Build package assembly
- Rebuild package Assembly

That way other assemblies (specially the uppsrc one) are not touched when doing a rebuild.

EDIT:
Unfortunately I couldn't find a method to find out what assembly a library belongs to.

EDIT:
Got it to work by adding the directory of the assembly to the package. I get this from the main Nest.

Rebuilding time got a lot lower

Adding the ability to build based on the color of the package. That way I can use a different color for different sections. Avoiding a rebuild on libraries that do not need to be rebuild.

---