Subject: Build a project composed by multiple shared libs
Posted by mdelfede on Fri, 17 Jan 2014 22:10:59 GMT
View Forum Message <> Reply to Message

Hi,

I'd like to build a set of shared libs at once. I'm doing it creating a main package that uses some tenths of packages, each of them building a shared lib :

MAIN
 |
 |-- lib1
 |-- lib2
 |-- .....
 |-- libn

I put the SO flag in main package, so all used packages are built as dll. The problem is in naming... in unix dlls should be named 'libXXXXX.so', but inside theide they're named 'MAIN_lib1.so'. If I rename and install the dlls, dependences between them is broken.
For example, if lib2 needs lib1, resulting shared libs are named MAIN_lib1.so and MAIN_lib2.so. If I rename them to liblib1.so and liblib2.so and install them, liblib2.so complains about missing MAIN_lib1.so.

Is there a possibility to force the name of built shared objects ? I know that it works for main package, adding a target, but not found a way for non-main packages.

Ciao

Max

Subject: Re: Build a project composed by multiple shared libs
Posted by mdelfede on Fri, 17 Jan 2014 22:54:17 GMT
View Forum Message <> Reply to Message

Hmmmm... the worse is that .so paths are hardcoded inside executable and between them. Is impossible to move them around.

Subject: Re: Build a project composed by multiple shared libs
Posted by dolik.rce on Sat, 18 Jan 2014 07:15:58 GMT
View Forum Message <> Reply to Message

Hi Max,

Would a scripting be possible workaround? You could have a main package for each of your libs, with proper target. Then you'd have a one "super main", that would include all of these packages and a macro file (something.usc). The supermain package doesn't need to be compiled, it would be just used to run the compilation of all the libs.

The macro would be quite simple, something like

```
macro "Build all libs" Ctrl+Alt+F5 {
    ClearConsole();
    // list of libs to build
    libs = [ "lib1", "lib2", lib3 ]
    for (p in libs) {
        // Get .upp file location for the package
        upp = PackageDir(p) + "/"+p+".upp";
        // Construct the target name
        target = "lib"+p+".so";
        // Build that package with currently set flags
        BuildProject(upp, Flags(), target);
    }
}
```

I know that it is not really a solution, but it could save you some time before a better way to do it is found

Best regards,
Honza

Subject: Re: Build a project composed by multiple shared libs
Posted by mdelfede on Sat, 18 Jan 2014 09:07:28 GMT
View Forum Message <> Reply to Message

Hi Honza,

thank you for suggestion.... it's not bad, but I guess it doesn't solve the dependencies between shared libs.
I mean.... what happen if lib2 depends on lib1 ?

Ciao

Max

Subject: Re: Build a project composed by multiple shared libs
Posted by dolik.rce on Sat, 18 Jan 2014 11:34:38 GMT
View Forum Message <> Reply to Message

mdelfede wrote on Sat, 18 January 2014 10:07what happen if lib2 depends on lib1 ?In that case you'd probably have to take care about it manually in the macro... Ensure the correct compilation order and copy the results to some directory from where the rest of the libs could access it.

So it is probably doable, but even less elegant than before

Honza

Subject: Re: Build a project composed by multiple shared libs
Posted by mdelfede on Sat, 18 Jan 2014 11:37:31 GMT
View Forum Message <> Reply to Message

Yep.... I think the best way would be t patch GccBuilder to generate libxxxx filenames and remove hardcoded paths inside .so modules....