Subject: Third-party lib packing HOWTO anybody?
Posted by GaroRobe on Tue, 17 May 2011 03:38:00 GMT

View Forum Message <> Reply to Message

Hi, %all%!

My situation is that I have to use some libs in my project (it's SDK for industrial cameras - in attachment). Now I would like to make a package containing headers and libs, making project easily distributable.

Click-it-all strategy doesn't help much (that's not a good thing BTW, but nevermind). Tried reading help and some topics around here but still can't get hang of mechanics.

Could you please explain me how do I make package I need?

```
File Attachments
1) CPP.rar, downloaded 477 times
```

Subject: Re: Third-party lib packing HOWTO anybody?
Posted by koldo on Tue, 17 May 2011 11:38:34 GMT

View Forum Message <> Reply to Message

GaroRobe wrote on Tue, 17 May 2011 05:38Hi, %all%!

My situation is that I have to use some libs in my project (it's SDK for industrial cameras - in attachment). Now I would like to make a package containing headers and libs, making project easily distributable.

Click-it-all strategy doesn't help much (that's not a good thing BTW, but nevermind). Tried reading help and some topics around here but still can't get hang of mechanics.

Could you please explain me how do I make package I need?

Hello Artem

Please include a very simple C/C++ console demo and I will return you a JAI folder and a JAI_Demo folder.

Subject: Re: Third-party lib packing HOWTO anybody?
Posted by GaroRobe on Wed, 18 May 2011 02:08:32 GMT

View Forum Message <> Reply to Message

Here - the very first by-the-book example. Doesn't compile - I know. But don't see why. Libs dir seems so be referenced... Well, that's the example anyway

Oh, right - I use VC10, VC10x64 and GCC (latest stable MinGW).

```
File Attachments
1) Test_1.rar, downloaded 471 times
```

Subject: Re: Third-party lib packing HOWTO anybody?
Posted by koldo on Wed, 18 May 2011 08:28:13 GMT
View Forum Message <> Reply to Message

Hello Artem

Here I enclose you the files. They compile for me with MSC9 and MinGW, all in 32 bits. As I do not have the dll, the program does not run.
Although 64 bits is implemented, I cannot test it.

You will see the changes, it is very easy :

- Folder Jai contains just the reference to the libs
- Fodler Jai_demo contains your demo with a few changes. Included compatibility with MinGW. Removed Eigen just to avoid any noise.
- Folder plugin_jai can be copied to uppsrc/plugin folder. .lib files have been renamed.

Please do not forget to include the include/lib/bin paths linking to plugin/jai/include, /lib and /bin folders.

```
File Attachments
1) jai_all.7z, downloaded 455 times
```

Subject: Re: Third-party lib packing HOWTO anybody?
Posted by GaroRobe on Thu, 19 May 2011 04:35:11 GMT
View Forum Message <> Reply to Message

That's weird... I copy jai from plugin_jai to uppsrc/plugin folder and get
Quote:LINK : fatal error LNK1104: cannot open file 'Jai_Factory32.lib'
Why? Somehow I don't get the way upp's building mechanism "thinks".

Subject: Re: Third-party lib packing HOWTO anybody?
Posted by koldo on Thu, 19 May 2011 06:29:01 GMT
View Forum Message <> Reply to Message

Hello Artem

Have you included this in addition to "Include" and "Path"? (last one for the .dll location):

Subject: Re: Third-party lib packing HOWTO anybody?
Posted by GaroRobe on Thu, 19 May 2011 06:39:40 GMT
View Forum Message <> Reply to Message

Aha - makes sence!

And is there any way to hard-code such requirements into packages? Like distribute libs in packages leaving only path requirements for the runtime?

Subject: Re: Third-party lib packing HOWTO anybody?
Posted by dolik.rce on Thu, 19 May 2011 08:04:42 GMT
View Forum Message <> Reply to Message

GaroRobe wrote on Thu, 19 May 2011 08:39Aha - makes sence!

And is there any way to hard-code such requirements into packages? Like distribute libs in packages leaving only path requirements for the runtime?
Yes, it can be done... altough it is not widely used You would have to specify the path in Project->Package organizer, adding a new link option (using context menu on right click in the right pane) and fill in something like -L../relative/path/to/lib/. The problem is that the syntax for GCC/MINGW is different than MSVC (don't know precise syntax), so you would have to add two separate entries, with when flags MSVC and !MSVC.

You could also use the very same way to directly add the lib into the linking command line. It might be simpler, but I am not sure if there aren't any hidden problems...

Best regards,
Honza

Subject: Re: Third-party lib packing HOWTO anybody?
Posted by GaroRobe on Thu, 19 May 2011 12:35:37 GMT
View Forum Message <> Reply to Message

"Not widely used"? And what's the downside of this approach?

Subject: Re: Third-party lib packing HOWTO anybody?
Posted by dolik.rce on Thu, 19 May 2011 14:31:12 GMT
View Forum Message <> Reply to Message

GaroRobe wrote on Thu, 19 May 2011 14:35"Not widely used"? And what's the downside of this approach?
The only downside is that you have to write it separately for each compiler, but that is not the reason why it is not used. The real reason is that it is rarely needed U++ is build around opensource libraries so they can be used in the form of source code, hence there is no need to distribute the libraries. Of course, I understand that sometime one has to use a proprietary library, but it is not very often. So that is why I said it is not widely used I believe that most of the programmers are lazy poeple, so when they find out they are using library that is not in path, they just add it in their global settings (in build methods, as Koldo showed you) and forget about it
Both solutions are valid, one is simple for you while the other one makes it simple for people to whom you distribute your packages.

Honza

Subject: Re: Third-party lib packing HOWTO anybody?
Posted by GaroRobe on Thu, 19 May 2011 23:22:53 GMT
View Forum Message <> Reply to Message

Okay now, guess I'm goin' a bit off the topic, but...

Could you please help me make this example run? It built without a prob now (a good thing), but crashes immidiately (a bad thing). Moreover, it crashes before a single step in debugger is performed and, half the times, drags TheIDE along to the nothingness (which is annoying).

Where should I start looking?

UPD: In x64 it actually complained not finding some dlls. Adding few from GenICam package made application launch, but I cannot debug it (because it is x64?).
The same trick with dlls didn't help with x86 version - still doesn't run.

Subject: Re: Third-party lib packing HOWTO anybody?
Posted by koldo on Fri, 20 May 2011 21:53:46 GMT
View Forum Message <> Reply to Message

Hello Artem

Please send all the files including dll to test it.