Subject: Small fixe for OPENBSD package organizer configuration
Posted by amrein on Wed, 22 Apr 2020 08:47:31 GMT
View Forum Message <> Reply to Message

Here is a small fix to get near to release state for OPENBSD (patch attached).

What I wanted to do (user cases) :

1. [DONE] - Get correct dependencies for X11, GTK, rt and dl on OPENBSD => small fixes for package organizer configuration is attached
2. [TODO] - Get INC += /usr/local/include and LIBPATH += /usr/local/lib inside package organizer configuration (but how?)

My only solution right now for the user case 2 is to add /usr/local/include and /usr/local/lib in a specific .bm file and call umk with this modified build method.
Is there a way to add a library path (LIBPATH) and include patch (INC) in package organizer configuration for a specific flag? According to U++ source code, LIBPATH and INC are pkg-config and .bm file dependent only.

File Attachments
1) OPENBSD_package_organizer_config_fix.patch, downloaded 364 times

Subject: Re: Small fixe for OPENBSD package organizer configuration
Posted by amrein on Wed, 22 Apr 2020 20:50:24 GMT
View Forum Message <> Reply to Message

I share this patch because I don't have enough svn write permission do fix those files myself (access denied).

Subject: Re: Small fixe for OPENBSD package organizer configuration
Posted by mirek on Thu, 23 Apr 2020 10:40:52 GMT
View Forum Message <> Reply to Message

I think that actually adding /usr/local/[lib, include] to default (I mean, autogenerated) .bm files is the trivial solution to the problem.

Do you see any downside?

Mirek

Subject: Re: Small fixe for OPENBSD package organizer configuration
Posted by amrein on Thu, 23 Apr 2020 18:43:31 GMT

View Forum Message <> Reply to Message

No downside but it's ugly?

Better to:

- add another array in Package class:
 Array<OptItem> library_path;

- add in PackageEditor::PackageEditor()
 Add("Internal include path", actual.include);
 Add("Internal library path", actual.library_path);

- Make actual.include like other options in Package Organizze (I mean, with rw access to 'When' textfield so it can be flag activated/distribution specific)

- Then use those Package variables to fill Makefiles and default .bm files if those flags are active

 ?

I didn't find where "Internal include" (package.include) is used in TheIDE, so I renamed it "Internal include path".

Subject: Re: Small fixe for OPENBSD package organizer configuration
Posted by mirek on Fri, 24 Apr 2020 06:04:42 GMT
View Forum Message <> Reply to Message

There should be no absolute paths listed in .upp files, that is the role of build methods (to act as "interface" between the source tree and build machine).

Now well, if these are autogenerated, maybe we can make it less ugly to detect OPENBSD when autogenerating them and add /usr/local just in that case?

(it would be better to detect in runtime, just in case that because of some miracle umks32 works, can we do that? What is the 'uname' output on OpenBSD?)

Subject: Re: Small fixe for OPENBSD package organizer configuration
Posted by mirek on Fri, 24 Apr 2020 07:05:25 GMT
View Forum Message <> Reply to Message

mirek wrote on Fri, 24 April 2020 08:04
Now well, if these are autogenerated, maybe we can make it less ugly to detect OPENBSD when autogenerating them and add /usr/local just in that case?

(it would be better to detect in runtime, just in case that because of some miracle umks32 works, can we do that? What is the 'uname' output on OpenBSD?)

Now in trunk - in OpenBSD, bm autogeneration should add /usr/local....

Also, patches are commited too. Next nightly should work with OpenBSD (I wonder whether there will more than zero users for that

Mirek

Subject: Re: Small fixe for OPENBSD package organizer configuration
Posted by amrein on Fri, 24 Apr 2020 09:51:48 GMT
View Forum Message <> Reply to Message

mirek wrote on Fri, 24 April 2020 08:04There should be no absolute paths listed in .upp files, that is the role of build methods (to act as "interface" between the source tree and build machine).

Now well, if these are autogenerated, maybe we can make it less ugly to detect OPENBSD when autogenerating them and add /usr/local just in that case?

(it would be better to detect in runtime, just in case that because of some miracle umks32 works, can we do that? What is the 'uname' output on OpenBSD?)

For me, adding internal include and internal libpath inside Package Organizer is the best way to have control over this without having to add hidden code to check for the OS and add /usr/local/include and /usr/local/lib if it's OpenBSD or any other new OS requiring this trick.

It's not user .bm includes list nor user .bm libpath list. It's an internal umk/ide/Core include and libpath and need to be added only for some OS like for instance here, OpenBSD.

Don't we already have include withing Package Organize (without the "When" clause)? I mean, if it already exist, why should we fight against it?

Subject: Re: Small fixe for OPENBSD package organizer configuration
Posted by mirek on Fri, 24 Apr 2020 12:07:54 GMT
View Forum Message <> Reply to Message

amrein wrote on Fri, 24 April 2020 11:51mirek wrote on Fri, 24 April 2020 08:04There should be no absolute paths listed in .upp files, that is the role of build methods (to act as "interface" between the source tree and build machine).

Now well, if these are autogenerated, maybe we can make it less ugly to detect OPENBSD when autogenerating them and add /usr/local just in that case?

(it would be better to detect in runtime, just in case that because of some miracle umks32 works, can we do that? What is the 'uname' output on OpenBSD?)

For me, adding internal include and internal libpath inside Package Organizer is the best way to have control over this without having to add hidden code to check for the OS and add /usr/local/include and /usr/local/lib if it's OpenBSD or any other new OS requiring this trick.

It's not user .bm includes list nor user .bm libpath list. It's an internal umk/ide/Core include and libpath and need to be added only for some OS like for instance here, OpenBSD.

Don't we already have include withing Package Organize (without the "When" clause)? I mean, if it already exist, why should we fight against it?

"need to be added only for some OS" is exactly the build method domain... This is the same as in Win32, where there is autodetection code for localizing C++ compiler folder and setting paths to it.

In either case, adding 4 lines of code to bm generator is much simpler than changing .upp structure, changing Package organizer GUI code and ide/Builders code....

Mirek

Subject: Re: Small fixe for OPENBSD package organizer configuration
Posted by amrein on Fri, 24 Apr 2020 17:40:08 GMT
View Forum Message <> Reply to Message

I guess, if it's only 4 lines for OpenBSD, it shouldn't hurt.

edit: It will only work on OpenBSD and not elsewhere. I mean, I will still have to use a modified .bm file to create correct Makefiles for umk or theide for OpenBSD if I'm on another platform. If not, I won't get correct INC and LIBPATH inside the Makefile. It's not a big issue.
