Subject: GCC.bm problems
Posted by ebojd on Thu, 22 Mar 2007 14:09:42 GMT
View Forum Message <> Reply to Message

There continues to be problems with the GCC.bm file missing from the source distribution. Looking at the structure of the file I surmise that it sets up various configuration *stuff*. I have the following questions and comments:

*) how is it generated?
*) when is it supposed to be generated?
*) how does it become "stale" or out of sync?

Having this break installation should be addressed IMNSHO...

EBo --

Subject: Re: GCC.bm problems
Posted by mirek on Thu, 22 Mar 2007 16:57:53 GMT
View Forum Message <> Reply to Message

ebojd wrote on Thu, 22 March 2007 10:09There continues to be problems with the GCC.bm file missing from the source distribution. Looking at the structure of the file I surmise that it sets up various configuration *stuff*. I have the following questions and comments:

*) how is it generated?
*) when is it supposed to be generated?
*) how does it become "stale" or out of sync?

Having this break installation should be addressed IMNSHO...

EBo --

Well, that file mostly consists of include/library paths and compiler options. It is user editable in "Build methods" dialog.

So far, I have created it long long time ago and then just copied the same file across various linux distros - it seems paths are pretty stable in fact.

Mirek

Subject: Re: GCC.bm problems
Posted by ebojd on Thu, 22 Mar 2007 17:10:17 GMT
View Forum Message <> Reply to Message

Thanks,

I figured as much. There are a few of the details which can/should be updated though (like the INCLUDE and LIB should either take the arguments from pkg-config or be initialized by them).

So to followup,

-> needs to be there on initial startup, but copying old is good enough.

-> basiclly does not become stale, but can be edited by user

The new revision of upp-portage (2007.1_rc3-r2) simply copies this over into the source tree. As soon as I sort out the problem with the new use initialization I'll post the code onto sourceforge.

EBo --

Subject: Re: GCC.bm problems
Posted by ebojd on Wed, 28 Mar 2007 15:42:47 GMT
View Forum Message <> Reply to Message

With the latest test I ran into a problem with the old definitions of GCC.bm. Specifically, it does not includ proper definitions on where to find the cflags and libs defined by pkg-config...

I propose to patch the Makefile to generate the GCC.bm file so that it will have the appropriate definitions at compile time.

Does this sound reasonable and acceptable?

EBo --

Subject: Re: GCC.bm problems
Posted by mirek on Wed, 28 Mar 2007 15:49:53 GMT
View Forum Message <> Reply to Message

But at least in thery, GCC.bm depends on actual target platform..

Mirek

Subject: Re: GCC.bm problems
Posted by ebojd on Wed, 28 Mar 2007 19:02:26 GMT
View Forum Message <> Reply to Message

Do we have a collection of what each of the variables are to be set for each of the platforms and compiler configurations? My comment regarding generating it as part of the compile/install phase in the Makefile is only valid for platforms which use the Makefile to do the initial build. If you give me a list of different configurations and settings I can probably get the Makefile to build it (either that or write a u++ app which queries the OS and configuration and builds one -- which would allow it to be regenerated on the fly with the one installed GCC.bm being the system default).

EBo --

Subject: Re: GCC.bm problems
Posted by mirek on Wed, 28 Mar 2007 20:46:39 GMT
View Forum Message <> Reply to Message

ebojd wrote on Wed, 28 March 2007 15:02Do we have a collection of what each of the variables are to be set for each of the platforms and compiler configurations? My comment regarding generating it as part of the compile/install phase in the Makefile is only valid for platforms which use the Makefile to do the initial build. If you give me a list of different configurations and settings I can probably get the Makefile to build it (either that or write a u++ app which queries the OS and configuration and builds one -- which would allow it to be regenerated on the fly with the one installed GCC.bm being the system default).

EBo --

Well, I am simply not sure this is worth the effort. Having existing .bm, variations are minimal and usually you can have it fixed in minutes. I think it will change two or three times on each distribution max. Fixing Makefile generator to generate .bm would be take days. Fixing generated Makefiles would not solve a problem, as you would have to fix Makefile more often then to fix .bm.

Mirek

Subject: Re: GCC.bm problems
Posted by ebojd on Thu, 29 Mar 2007 12:54:38 GMT
View Forum Message <> Reply to Message

luzr wrote on Wed, 28 March 2007 15:46
Well, I am simply not sure this is worth the effort. Having existing .bm, variations are minimal and usually you can have it fixed in minutes. I think it will change two or three times on each distribution max. Fixing Makefile generator to generate .bm would be take days. Fixing generated Makefiles would not solve a problem, as you would have to fix Makefile more often then to fix .bm.

Mirek

OK. My thought was that without it, it does not just "run out of the box". Once you have anything

running, then I agree that the changes are trivial. The real problem comes for someone building u++ for the very first time, and has no clue why it is broken... When this happens to me I first ask myself why it does not work, then I ask myself if it is worth trying to figure out what went wrong, and lastly I ask myself if the basic install fails, then what else did they get wrong. U++ is a really great tool, and having an install break out of the box leaves the wrong impression IMHO.

EBo --

Subject: Re: GCC.bm problems
Posted by mirek on Thu, 29 Mar 2007 13:00:41 GMT
View Forum Message <> Reply to Message

ebojd wrote on Thu, 29 March 2007 08:54luzr wrote on Wed, 28 March 2007 15:46
Well, I am simply not sure this is worth the effort. Having existing .bm, variations are minimal and usually you can have it fixed in minutes. I think it will change two or three times on each distribution max. Fixing Makefile generator to generate .bm would be take days. Fixing generated Makefiles would not solve a problem, as you would have to fix Makefile more often then to fix .bm.

Mirek

OK. My thought was that without it, it does not just "run out of the box". Once you have anything running, then I agree that the changes are trivial. The real problem comes for someone building u++ for the very first time, and has no clue why it is broken... When this happens to me I first ask myself why it does not work, then I ask myself if it is worth trying to figure out what went wrong, and lastly I ask myself if the basic install fails, then what else did they get wrong. U++ is a really great tool, and having an install break out of the box leaves the wrong impression IMHO.

EBo --

Well, I think we should rather try for per-platform presence (.deb, .rpm etc.)

Mirek

Subject: Re: GCC.bm problems
Posted by ebojd on Thu, 29 Mar 2007 15:42:22 GMT
View Forum Message <> Reply to Message

luzr wrote on Thu, 29 March 2007 08:00
Well, I think we should rather try for per-platform presence (.deb, .rpm etc.)

Mirek

OK. I'll keep my specific changes in the ebuild then.

Cheers,

EBo --
