

---

Subject: What is the purpose of "GCC.bm.in" in tarbals?

Posted by [Klugier](#) on Sun, 05 Oct 2014 13:09:12 GMT

[View Forum Message](#) <> [Reply to Message](#)

---

Hello,

I can not fully understand what is the purpose of file "GCC.bm.in". The first problem I have found is when you copy this file to "~/.upp/theide" the build method will be not detected. At least until the time when you change the name to "GCC.bm". Moreover even "Build methods" (Ide -> Setup -> Build methods..) cannot import this file. It can only works with "GCC.bm". Simply rename do the trick.

Do we need "GCC.bm.in" which is the same file as "GCC.in", but it posses import limitation?

P.S. 1

"GCC.bm" is not included in nightly tarbals. What is interesting here is that makefile can react on succesfull ide build and copy "GCC.bm" to following directory "~/.upp/theide" (Do not override if the file exists!!!). This small change will guarantee that IDE will works out of the box on POSIX.

P.S. 2

IDE should have capability to export certain build method or all methods.

Sincerely,

Klugier

---

---

Subject: Re: What is the purpose of "GCC.bm.in" in tarbals?

Posted by [dolik.rce](#) on Mon, 06 Oct 2014 06:40:28 GMT

[View Forum Message](#) <> [Reply to Message](#)

---

Hi Klugier

Klugier wrote on Sun, 05 October 2014 15:09Hello,

I can not fully understand what is the purpose of file "GCC.bm.in". The first problem I have found is when you copy this file to "~/.upp/theide" the build method will be not detected. At least until the time when you change the name to "GCC.bm". Moreover even "Build methods" (Ide -> Setup -> Build methods..) cannot import this file. It can only works with "GCC.bm". Simply rename do the trick.

Do we need "GCC.bm.in" which is the same file as "GCC.in", but it posses import limitation?

P.S. 1

"GCC.bm" is not included in nightly tarbals. What is interesting here is that makefile can react on succesfull ide build and copy "GCC.bm" to following directory "~/.upp/theide" (Do not override if the file exists!!!). This small change will guarantee that IDE will works out of the box on POSIX. Files called "\*.in" are commonly used in Makefiles and other build systems as templates. They do

not contain fully functional content, but they are used as input to the build system, which fills in the missing parts to create full file. This is mostly used to allow different files on different systems, the build system always fill correct values. In this particular case, it takes care of the library paths, which can be different between various Linux distributions.

Klugier wrote on Sun, 05 October 2014 15:09

P.S. 2

IDE should have capability to export certain build method or all methods.

Why? It seems perfectly fine for me to just copy the file \*.bm from ~/.upp/ide if I need it to move somewhere else. It is actually simpler then exporting it, because I don't have to start TheIDE first

Best regards,

Honza

---

---

Subject: Re: What is the purpose of "GCC.bm.in" in tarbals?

Posted by [Klugier](#) on Mon, 06 Oct 2014 16:57:46 GMT

[View Forum Message](#) <> [Reply to Message](#)

---

Hello dolik.rce,

The main problem with "GCC.bm.in" is that it cannot be used as build method. Moreover tarbal dosen't offer any build method (.bm file), so after successful compilation, ide cannot compile any application. This is huge drawback, because you will need to configure ide that you dosen't know.

Like I sad before we can copy "GCC.bm" to "~/.upp/ide" after build and then user can use ide out of the box.

GCC.bm IS THE SAME FILE AS GCC.bm.in !!!

Quote:

Why? It seems perfectly fine for me to just copy the file \*.bm from ~/.upp/ide if I need it to move somewhere else. It is actually simpler then exporting it, because I don't have to start TheIDE first  
Smile

Fine, but new user dosen't know where ide stores its build methods. Moreover "~/.upp/ide" is hard to find. I have discovered this directory after one year of using Upp. So the idea of adding export beside import option seems to be reasonable.

Sincerely,

Klugier

---

---

Subject: Re: What is the purpose of "GCC.bm.in" in tarbals?

Posted by [dolik.rce](#) on Tue, 07 Oct 2014 04:27:39 GMT

[View Forum Message](#) <> [Reply to Message](#)

---

Klugier wrote on Mon, 06 October 2014 18:57The main problem with "GCC.bm.in" is that it cannot be used as build method. Moreover tarbal dosen't offer any build method (.bm file), so after successful compilation, ide cannot compile any application. This is huge drawback, because you will need to configure ide that you dosen't know. It is not intended to be used as build method. It is only template, the actual build method is created when building TheIDE.

Klugier wrote on Mon, 06 October 2014 18:57Like I sad before we can copy "GCC.bm" to "~/.upp/ide" after build and then user can use ide out of the box.GCC.bm is installed automatically, when you call either "make install" or the "doinstall" script directly.

Klugier wrote on Mon, 06 October 2014 18:57GCC.bm IS THE SAME FILE AS GCC.bm.in !!!No, it is not. The GCC.bm.in contains placeholder "((INCLUDES))". This string is replaced by "domake" script, to create actuall GCC.bm, using this exact code:

```
if which pkg-config
then
  sed -e "s@-I((INCLUDES))@`pkg-config --cflags-only-I gtk+-2.0` @g" uppsrc/Makefile.in
>uppsrc/Makefile
  sed -e "s@((INCLUDES))@`pkg-config --cflags-only-I gtk+-2.0|sed -e s/-I//g -e \"s/ /;/g\" @g"
GCC.bm.in >GCC.bm
else
  sed -e "s@((INCLUDES))@ @g" uppsrc/Makefile.in >uppsrc/Makefile
  sed -e "s@((INCLUDES))@ @g" GCC.bm.in >GCC.bm
fi
```

That means, that if there is pkg-config installed on the target machine, it will be used to determine correct include paths for gtk+-2.0, which is known to vary between its versions.

Klugier wrote on Mon, 06 October 2014 18:57Quote:

Why? It seems perfectly fine for me to just copy the file \*.bm from ~/.upp/ide if I need it to move somewhere else. It is actually simpler then exporting it, because I don't have to start TheIDE first Smile

Fine, but new user dosen't know where ide stores its build methods. Moreover "~/.upp/ide" is hard to find. I have discovered this directory after one year of using Upp. So the idea of adding export beside import option seems to be reasonable.

I'm not saying I'm strictly opposed to the idea. Just that it seems redundant to me. It could be also fixed by providing better documentation. Actually, it is even documented in the Getting started with TheIDE article, there is only slight mistake in the directory path.

Honza

---