
Subject: Problem with hiding (shadowing) variables

Posted by [Mohammad Al-Turany](#) on Fri, 18 Sep 2009 22:01:53 GMT

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

Hello everybody,

I noticed that there is a lot of places in PandaRoot where variables are simply shadowed in the software, at some places this is harmless but it could also create serious problems which are hard to find out! that is why I switch on the warning for hiding variables for the compiler (-Wshadow) this will show you the places where this happened, So please try to avoid this and correct the corresponding classes and by the way many of these could be avoided if you simply try to follow the coding convention which we agreed upon years ago!

<http://panda-wiki.gsi.de/cgi-bin/view/Computing/PandaRootCodingRules>

regards

Mohammad

Subject: Hiding variables... take them seriously!!!

Posted by [Johan Messchendorp](#) on Mon, 28 Sep 2009 20:51:48 GMT

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

Dear developers,

More than a week ago, Mohammad switched on the -Wshadow flag during the compilation of PandaRoot. As a result, plenty of potential problems were revealed. Some of the developers started already correcting these warnings by following the coding conventions as we agreed upon more than a year ago. Nevertheless, there are still many (really many, more than 1500!!) warnings showing up during compilation. So please, correct these ASAP. It is better to do it now, then in a later stage!! If you have any questions related to this, do not hesitate to ask.....

Thanks in advance,

Johan.

Subject: Re: Hiding variables... take them seriously!!!

Posted by [Sebastian Neubert](#) on Tue, 29 Sep 2009 09:15:27 GMT

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

Hey guys!

Here I post what Linus Torvalds has to say about -Wshadow (from <http://kerneltrap.org/node/7434>). Please consider removing this warning.

Quote:

On Wed, 29 Nov 2006, Jesper Juhl wrote:

>

> I would venture that "-Wshadow" is another one of those.

I'd agree, except for the fact that gcc does a horribly `_bad_` job of -Wshadow, making it (again) totally unusable.

For example, it's often entirely interesting to hear about local variables that shadow each other. No question about it.

HOWEVER. It's `_not_` really interesting to hear about a local variable that happens to have a common name that is also shared by an extern function.

There just isn't any room for confusion, and it's actually not even that unusual - I tried using -Wshadow on real programs, and it was just horribly irritating.

In the kernel, we had obvious things like local use of "jiffies" that just make `_total_` sense in a small inline function, and the fact that there happens to be an extern declaration for "jiffies" just isn't very interesting.

Similarly, with nested macro expansion, even the "local variable shadows another local variable" case - that looks like it should have an obvious warning on the face of it - really isn't always necessarily that interesting after all. Maybe it is a bug, maybe it isn't, but it's no longer `_obviously_` bogus any more.

So I'm not convinced about the usefulness of "-Wshadow". ESPECIALLY the way that gcc implements it, it's almost totally useless in real life.

For example, I tried it on "git" one time, and this is a perfect example of why "-Wshadow" is totally broken:

```
diff-delta.c: In function 'create_delta_index':
diff-delta.c:142: warning: declaration of 'index' shadows a global declaration
```

(and there's a `_lot_` of those). If I'm not allowed to use "index" as a local variable and include `<string.h>` at the same time, something is simply SERIOUSLY WRONG with the warning.

So the fact is, the C language has scoping rules for a reason. Can you screw yourself by using them badly? Sure. But that does NOT mean that the same name in different scopes is a bad thing that should be warned about.

If I wanted a language that didn't allow me to do anything wrong, I'd be using Pascal. As it is, it turns out that things that "look" wrong on a local level are often not wrong after all.

Linus

Subject: Re: Hiding variables... take them seriously!!!
Posted by [Stefano Spataro](#) on Tue, 29 Sep 2009 10:05:05 GMT
[View Forum Message](#) <> [Reply to Message](#)

If you know what you are doing, then probably all those warnings are useless. But when you call a data member in your class "x", and then inside your class you define in each function new variables called "x", then are you really sure that you are always using the proper "x"? If you forget one time to define again this x, the compiler does not help you and you will use the wrong one.

One case is inside emc package, where the code was simply taken from the other framework, and there are many of these warnings. Using our coding rules, such as calling the data members fX as decided ages ago, one can avoid those warnings.

Therefore, I think these warnings are a good excuse to recheck the code being sure on whay it is really doing.

Subject: Re: Hiding variables... take them seriously!!!
Posted by [Sebastian Neubert](#) on Tue, 29 Sep 2009 10:11:30 GMT
[View Forum Message](#) <> [Reply to Message](#)

Hi Stefano!

You are right with your concerns. These kinds of things have to be respected. The point is that gcc does not really help you here because it gives too many warnings that come not from the case you described.

Cheers! Sebastian.

Subject: Re: Hiding variables... take them seriously!!!
Posted by [Mohammad Al-Turany](#) on Tue, 29 Sep 2009 10:21:09 GMT
[View Forum Message](#) <> [Reply to Message](#)

Hi,

As I use this flag there was a reason for it and not because I have nothing to do or I searched in google what people think about it! and the guy you quote here also do not like the warning about comparing signed with unsigned integers, for him this is also silly! (fortunately this guy wrote only 2% of the linux kernel)

If I get the message properly, people have better things to do than removing such stupid warnings or following stupid coding conventions, and if somebody like Linus said it is not important then it is not important! we do not need to understand anything or even look for it, Linus said it is not important!

Anyway, some reasons which linus did not talk about are:

1. Some people redefine TrackId, EventId, etc locally in some functions so what does the experts think about this, is it harmless!
2. Re-define of base class variables, is it also harmless! and it get more harmless when the people redefine some objects (TVector3) in the stepping so it is created and deleted also for each step!
3. Class variables, global variables and local ones with the same name, the compiler know how to distinguish between them, but how readable is this code for other people!

So because of all this we will not remove this warning unless every body in panda thinks the same like you and Linus, then we will switch it off only for PANDA! other people agreed and want to have it and they think that they are clever enough to decide for themselves if these warning has a meaning or not!

Mohammad

Subject: Re: Hiding variables... take them seriously!!!
Posted by [Sebastian Neubert](#) on Tue, 29 Sep 2009 11:16:47 GMT
[View Forum Message](#) <> [Reply to Message](#)

Dear Mohammad,

Linus did not only state an opinion but he offered an explanation why there is a controversy in the community about this warning.

I agree with you that people have to be careful about all the things that can go wrong with redefinition of symbols. If the warning would work the way you intend to use it, things would be great. Apparently this is unfortunately not the case and actually seems to prevent readable code in a lot of circumstances.

In fact people do have better things to do.

Best Regards, Sebastian.

Subject: Re: Hiding variables... take them seriously!!!
Posted by [Mohammad Al-Turany](#) on Tue, 29 Sep 2009 11:36:21 GMT
[View Forum Message](#) <> [Reply to Message](#)

Hallo Sebastian,

May be you can give us an example from PandaRoot where this warning is not showing it

should show? Or where there is no shadowing but it shows one?

Mohammad

Subject: Re: Hiding variables... take them seriously!!!
Posted by [Sebastian Neubert](#) on Tue, 29 Sep 2009 12:34:06 GMT
[View Forum Message](#) <> [Reply to Message](#)

Dear Mohammad,

here is one example:

```
simulation/trunk/tpc/PndTpcGas.h: In member function 'double PndTpcGas::VDrift(double, double) const':  
simulation/trunk/tpc/PndTpcGas.h:52: warning: declaration of 'B' shadows a member of 'this'  
simulation/trunk/tpc/PndTpcGas.h:52: warning: declaration of 'E' shadows a member of 'this'
```

Indeed the member variables are prefixed by _ which is for some reason not recognized when the warning is created. The prefix is a leftover when I ported my work form the older frameworks.

Sebastian.

Subject: Re: Hiding variables... take them seriously!!!
Posted by [Mohammad Al-Turany](#) on Tue, 29 Sep 2009 13:24:04 GMT
[View Forum Message](#) <> [Reply to Message](#)

Hallo Sebastian,

it good that you give this example! in fact what you have here is also some kind of violation against the C++ standard,

Here's what the C++ Standard says (17.4.3.1.2):

Quote: "Each name that contains a double underscore (_ _) or begins with an underscore followed by an uppercase letter is reserved to the implementation"

However, some compilers allow this, because for all it knows your code could be part of the OS or libc that has the legitimate right to use such identifiers. But technically it's actually undefined behavior, the reason being that you may tread on a reserved name.

In our case here the compiler simply ignore your underscore and then you get this warning. So it is not completely meaningless and you can look yourself that many books and forums advice not use the underscore in a variable name!

regards

Mohammad

Subject: Re: Hiding variables... take them seriously!!!
Posted by [Sebastian Neubert](#) on Tue, 29 Sep 2009 14:43:14 GMT
[View Forum Message](#) <> [Reply to Message](#)

Hi Mohammad,

I have just renamed all member variables in the tpc package (not yet tpcresco) to conform with the prefixing with f instead of _ in revision 6629.

I still get a lot of warnings. As far as I can see most of them occur in a situation like this:

```
const TMatrixD& cov() const {return fcov;}  
....  
....  
void SetCov(const TMatrixD& cov){fcov=cov;fhasaxis=false;}
```

I still believe this is acceptable code. Removing all these situations would take a bit more time.

Regards, Sebastian.

PS: For future reference I used this:
for i in *.{h,cxx} ; do sed -i.bak -e 's/(\W)_/1f/g' \$i; done

Subject: Re: Hiding variables... take them seriously!!!
Posted by [Mohammad Al-Turany](#) on Tue, 29 Sep 2009 14:56:37 GMT
[View Forum Message](#) <> [Reply to Message](#)

Hi Sebastian,

it seems that you forget to check in the "PndTpcRiemannHough" the cmake crashes with:

Quote:CMake Error in tpc/tpcresco/CMakeLists.txt:
Cannot find source file "PndTpcRiemannHough.cxx". Tried extensions .c .C
.c++ .cc .cpp .cxx .m .M .mm .h .hh .h++ .hm .hpp .hxx .in .txx

Mohammad

Subject: Re: Hiding variables... take them seriously!!!
Posted by [Sebastian Neubert](#) on Tue, 29 Sep 2009 15:05:28 GMT
[View Forum Message](#) <> [Reply to Message](#)

Hi Mohammad!

I added this class. Please try again.

Regards, Sebastian.

Subject: Re: Hiding variables... take them seriously!!!
Posted by [Mohammad Al-Turany](#) on Wed, 30 Sep 2009 09:59:33 GMT
[View Forum Message](#) <> [Reply to Message](#)

Hi,

Yes, now it compiles but still another problem when loading the TPC library, it seems that your replacement script (command) also replace a line in the tpcLinkDef.h

```
#ifdef __CINT__ with #ifdef f_CINT__
```

which compile but failed to load the library, which crashes all test macros:

<http://fairroot.gsi.de/CDash/index.php?project=PandaRoot&date=>

anyway I correct it now in SVN.

Mohammad

Subject: Re: Hiding variables... take them seriously!!!
Posted by [Stefano Spataro](#) on Thu, 29 Oct 2009 16:33:08 GMT
[View Forum Message](#) <> [Reply to Message](#)

Hi,

I have fixed the shadowing for tpc code.

I have run some events and a reconstruction chain, obtaining "decent" results.

If somebody will find somethign strange, please mail me and I will check if I have introduced some error (even if my changes should not affect at all the calculations).

There are still two strange warnings in tpc code. Sebastian is informed about them.

Subject: Re: Hiding variables... take them seriously!!!
Posted by [Johan Messchendorp](#) on Thu, 29 Oct 2009 21:46:17 GMT
[View Forum Message](#) <> [Reply to Message](#)

Thx Stefano!

Johan.
