

---

Subject: Problem when using boost libraries

Posted by [Stanislav Poslavsky](#) on Fri, 25 Oct 2013 12:56:28 GMT

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

---

Good day.

I'm working under implementation of a custom generator in PandaRoot and would like to use some boost libraries. All ok, when I use boost::random, but, when I simply try to include

```
#include <boost/date_time/posix_time/posix_time.hpp>
```

I have a very long stack of compilation errors with a typical line looking as

```
Error: G__getvariable: expression
```

```
/home/stas/distr/pandaroot/external/build//include/boost/config.hpp:23:
```

Same with other boost libraries, like boost::iostreams etc. After some search with google, I found that problems might arise from incompatibilities of boost and CINT preprocessors.

So, is there any way to use boost libraries inside PandaRoot?

Thanks, Stanislav.

---

---

Subject: Re: Problem when using boost libraries

Posted by [Mohammad Al-Turany](#) on Fri, 25 Oct 2013 13:15:39 GMT

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

---

Hi,

in your header file:

```
#if !defined(__CINT__) || defined(__MAKECINT__)  
#include <boost/date_time/posix_time/posix_time.hpp>  
#endif
```

Cheers,

Mohammad

---

---

Subject: Re: Problem when using boost libraries

Posted by [Stanislav Poslavsky](#) on Fri, 25 Oct 2013 13:26:16 GMT

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

---

Thanks!

But what if this condition fails (`\_\_CINT\_\_` is defined and `\_\_MAKECINT\_\_` is not) => #include statement is ignored? How I can use boost libraries in this case?

(I see only one way - compile my library without \_\_CINT\_\_ and make a .so file, then use it as an external complied library - but, I try to find another solution)

---

---

Subject: Re: Problem when using boost libraries

Posted by [Mohammad Al-Turany](#) on Fri, 25 Oct 2013 13:36:14 GMT

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

---

These are the only conditions which are used in ROOT when CINT is processing your class or create a dictionary for it. You can also put the boost includes only in your implementation files to avoid this! in any case in the near future will get rid of CINT completely!

Cheers,  
Mohammad

---

---

Subject: Re: Problem when using boost libraries

Posted by [Stanislav Poslavsky](#) on Fri, 25 Oct 2013 13:41:37 GMT

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

---

It works! When I put #include only in .cxx all goes fine.

Many thanks,  
Stanislav

---