

CoCoA - Bug #57

Bug # 53 (Closed): MSVC10 compilation

MSVC10 - BOOST_SCOPE_EXIT

07 Dec 2011 09:42 - Anna Maria Bigatti

Status:	Closed	Start date:	07 Dec 2011
Priority:	Normal	Due date:	
Assignee:		% Done:	100%
Category:	Portability	Estimated time:	0.00 hour
Target version:	CoCoA-5.0.2	Spent time:	2.20 hours
Description			
With Visual Studio there are problems with BOOST_SCOPE_EXIT and this alternative is proposed (thanks to Bruno Simoes):			
<pre>#define LIGHT_SCOPE_EXIT \ struct BOOST_PP_CAT (local_struct_, __LINE__) { \ BOOST_PP_CAT (~local_struct_, __LINE__) () +#define LIGHT_SCOPE_EXIT_END \ } BOOST_PP_CAT (local_struct_var_, __LINE__); static_cast<void> (BOOST_PP_CAT (local_struct_var_, __LINE__));</pre>			
Is that OK?			

History

#1 - 07 Dec 2011 09:43 - Anna Maria Bigatti

- Category set to Portability

#2 - 07 Dec 2011 09:43 - Anna Maria Bigatti

- % Done changed from 0 to 30

#3 - 07 Dec 2011 10:01 - Anna Maria Bigatti

- Subject changed from MSVC10 compilation - BOOST_SCOPE_EXIT to MSVC10 - BOOST_SCOPE_EXIT

#4 - 22 Mar 2012 16:05 - John Abbott

I wonder why the only reports I can find on the internet about problems with BOOST_SCOPE_EXIT and Visual C++ are ours. Why does no one else have any problems?

Which version of BOOST was Bruno using?

There's something odd going on here. Maybe we are using BOOST_SCOPE_EXIT wrongly?

#5 - 03 Apr 2012 18:50 - John Abbott

- Status changed from New to Closed

- % Done changed from 30 to 100

The problem was that BOOST_SCOPE_EXIT requires at least 1 arg, though this is not entirely evident from their documentation. Someone has already registered a request that BOOST modify their impl to accept no argument.

JAA modified the code by passing (&interpreter) as argument -- not strictly necessary because interpreter is a global [naughty Giovanni].

We have confirmed that the modified code compiles fine on Linux, MacOSX and Windows.

#6 - 13 Jul 2012 15:51 - Anna Maria Bigatti

- *Target version set to CoCoA-5.0.2*