

CoCoALib - Feature #639

Shadow CoCoA namespace to help guarantee portability (without ambiguity)

28 Oct 2014 15:21 - John Abbott

Status:	New	Start date:	28 Oct 2014
Priority:	Low	Due date:	
Assignee:		% Done:	0%
Category:	Safety	Estimated time:	0.00 hour
Target version:	CoCoALib-1.0	Spent time:	0.75 hour
Description			
<p>Some problems of ambiguity in the CoCoALib source code have recently arisen. They have appeared because recent versions of C++ have introduced functions in the <code>::std</code> namespace with the same name as functions in the CoCoA namespace.</p> <p>To help avoid future problems of ambiguity suddenly appearing we could, for each function in the CoCoA, define an (empty) function in a shadow CoCoA namespace having the same name. If our CoCoALib code is not ambiguous when the shadow namespace is visible then I think ambiguity cannot arise in the CoCoALib code whatever functions may be added to the <code>::std</code> namespace.</p> <p>Action: implement a "shadow" CoCoA namespace with declarations of functions (and classes?) having names the same as those in CoCoALib.</p>			
Related issues:			
Related to CoCoALib - Design #619: Modulus (for CRTMill) ambiguous		Closed	09 Sep 2014
Related to CoCoALib - Bug #631: Ambiguous: rank for matrix (in ex-matrix1.C)		Closed	26 Sep 2014

History

#1 - 28 Oct 2014 15:26 - John Abbott

- Priority changed from Normal to Low

I don't know if this would really be worth the trouble.

I'm not even entirely sure that it would give the guarantees I want: in issue [#619](#) the problem was a template class whose name coincided with a function name in CoCoALib (so not simply the presence of another homonymous function in another namespace).

#2 - 28 Oct 2014 15:34 - John Abbott

Perhaps it is not so crucial to do this; or, at least, there may be a good compromise.

It is irritating when an ambiguity pops up, but an ambiguity does not lead to incorrect program behaviour (a grave problem). Of course, new ambiguities will trigger source code maintenance.

Perhaps we could create the shadow namespace containing just copies of whatever was in `::std` which caused the problem; this could be useful for developers using old C++ compilers (and old versions of the STL). This would be far less onerous than adding "fake" declarations of every CoCoALib function, and probably just as useful...