X3J16/96–0066
WG21/N0884
March 12, 1996
Andrew Koenig, AT&T Research

Response to 96-0057/N0875 (independent compilation of templates)

Overview

Separate compilation of templates has been part of C++ since before the ARM was published. In its November 1994 meeting, the committee reaffirmed separate compilation; that reaffirmation appeared in the May 1995 CD.

John Spicer submitted a proposal in the pre-Santa Cruz mailing to remove separate compilation for templates. At the Santa Cruz meeting, he resubmitted that proposal, which was changed only by the addition of a list of "sponsors." That proposal raises several objections to separate compilation, which the Core3 working group discussed at length on March 11 and the committee discussed in a technical session during the evening of March 11. Based on the discussions in that working group meeting and elsewhere, I believe that the objections do not hold up. The reasons follow.

Clarity of description

The strongest claim is that separate compilation has so many problems in its description that it is not clear whether they can be resolved in time to produce a good standard. Further examination, however, showed that almost all of the problems remain even in the absence of separate compilation, because they show up when combining templates and namespaces. Thus removing separate compilation does not significantly reduce the committee's workload.

Ease of implementation

Spicer's proposal claims that the new model has never been implemented. However, two working group members said they have implemented it. At least one of those implementations is likely to become commercially available. In subsequent discussion, John Spicer said that he thought separate compilation was implementable.

Efficiency

The proposal also claims that separate compilation "cannot be implemented efficiently enough to be usable." That is surely a matter of opinion, and one about which users should be allowed to make their own tradeoffs. Separate compilation is intrinsically more efficient for large systems because it requires only template declarations to be reprocessed for each translation unit and not their definitions also.

Ease of use

The paper claims that separate compilation is hard to use. Again, this is a matter of opinion that users should decide for themselves. Separate compilation is not particularly difficult for non-template programs to use, and I have seen no evidence that templates introduce any new difficulties for real users writing real programs. Moreover, not having separate compilation introduces a major pitfall: macros that users define in their programs might potentially conflict with any name used in a template implementation, including names of local variables. How can we justify saying that templates had better not have a local variable named **j0** because some C++ implementations define a macro named **j0** in **<math.h>**? See Dag Brück's paper (N0882/96–0064) for more details.

Summary

John Spicer's proposal asks the committee to take an unprecedented action: remove from the second CD a major feature that was part of the first CD. Extraordinary actions require extraordinary justification. The arguments in the proposal, while strongly stated, do not hold up under scrutiny. I hope the committee will have the good judgment to vote down this proposal and put the matter to rest.