

DCMTK - Feature #172

Spezialisierung von OFList für Pointertypen

2002-06-21 00:00 - Marco Eichelberg

Status: Rejected Priority: Normal Assignee: Category: Target version: Module: ofstd Operating System:	Start date: Due date: % Done: 0% Estimated time: 0:00 hour Compiler:
Description OFList wird häufig verwendet, um Pointertypen zu verwalten. Man könnte den DCMTK-Binärcode deutlich verkleinern, wenn man <code>* OFList<void *></code> explizit expandiert (in einer .cc-Datei) <code>*</code> eine partielle Spezialisierung <code>OFList<class T*></code> realisiert, die inline auf <code>OFList<void *></code> castet. Da partielle Spezialisierung von den meisten Compilern noch nicht unterstützt wird, müsste man das ganze explizit machen: <pre>typedef OFList<void *> OFVoidPointerList;</pre><pre>OFPointerList<class T *>: public (private?) OFVoidPointerList { inline name(T* x) { Pointerlist::name((void *)x) } }</pre> Name: <pre>OFPtrList OFPList OFPointerList<class T *></pre> Auch die Iteratoren müßten entsprechend angepaßt werden.	

History

#1 - 2025-08-29 17:57 - Marco Eichelberg

- Status changed from New to Rejected

This might cause issues when STL `std::list` is in use, and a better way to reduce binary size is to use shared libraries, which is meanwhile officially supported.