C++ Conversion BoF

GNU Cauldron 2012
Prague

wiki/cxxx-conversion
Purpose

Enable modern coding techniques.

Encourage better decomposition.

Ease the task of coming up to speed in GCC.

Provide better type safety.

Method: Upgrade infrastructure to make
Philosophy

Each change has a benefit. One or more of:

- Faster compilation.
- Smaller compiler.
- More concise source.
- Better type safety.
- More source flexibility.
Process

1. Prove
   ○ Change an existing abstraction to C++.
   ○ Convert a few uses to the new syntax.
2. Commit
   ○ Push to trunk.
3. Expand
   ○ Convert other uses to the new syntax.
4. Exploit
   ○ Consider representation improvements.
Coding Conventions

C++03 in gcc, libc++ and fixincludes.

Much remains the same.

C++ conventions are conservative.
  ○ No exceptions or RTTI.
  ○ Be reasonable and justifiable.
  ○ Declare variables at first use.
  ○ Avoid implicit conversions.
  ○ Take care in overloading functions.
  ○ Define member functions out-of-class.
Challenge: Garbage Collection

GGC does not understand C++.

GTY marking is a burden.

Marking stdlib is not desirable.

GGC required for PCH.

C++ can simplify GGC code.
Challenge: ICE Report

Macros provide file/line information.

Replacing macro calls with C++ operators eliminates that information.

Instead dump (part of) stack on ICE.

Mechanism and format TBD.
Status

Bootstrap switched to C++.  
Vector converted to C++. Still needs []).  
Type safe hash table in use.  
Gengtype hacked for vectors.  
Tree check macros replaced with debuggable inline functions.  
Coding conventions patch submitted.
Possible Conversion Tasks

Operators for double_int, vector.
Scoped timevars, pointers.
Classes for tree_list, cgraph, tree.
Strongly typed cgraph, tree pointers.
Accessor methods rather than macros.
Member operator new.

....