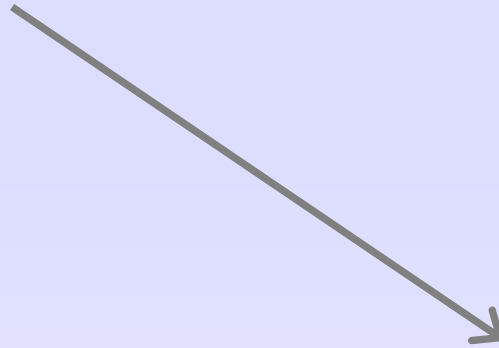


郭靖

武艺精湛，是武林大豪杰，一生为国为民献身。



Precompiled Headers In Action



Translates to:

“I, Precompiled-Header-san,
chop your compile time to pieces!”

Application Area • 2006-09-27

Axel Naumann, PH/SFT

Do We Care?

Not just theory: enabled by default for ROOT
on GCC4, Windows MSVC; ICC ≥ 8

Real build time speed improvement

Easy to use once you understand how

Goal of this talk: explain what PCH are, why
you want to use them, and how to use them.

Compiler & Headers

#include = copy & paste headers into sources:

Header.h: #inc Header1.h #inc Header2.h #inc Header3.h #inc Header4.h #inc Header5.h #inc Header6.h #inc Header7.h #inc Header8.h #inc Header9.h	Header.h: #inc Header1.h #inc Header2.h #inc Header3.h #inc Header4.h #inc Header5.h #inc Header6.h #inc Header7.h #inc Header8.h #inc Header9.h	Header.h: #inc Header1.h #inc Header2.h #inc Header3.h #inc Header4.h #inc Header5.h #inc Header6.h #inc Header7.h #inc Header8.h #inc Header9.h	Header.h: #inc Header1.h #inc Header2.h #inc Header3.h #inc Header4.h #inc Header5.h #inc Header6.h #inc Header7.h #inc Header8.h #inc Header9.h
Source1.cxx: #inc Header.h	Source2.cxx: #inc Header.h	Source3.cxx: #inc Header.h	Source4.cxx: #inc Header.h

The compiler compiles everything that's white

Compiler & Precompiled Headers

#include of a precompiled header: first compile header, then sources that include it

Header.h:

```
#inc Header1.h  
#inc Header2.h  
#inc Header3.h  
#inc Header4.h  
#inc Header5.h  
#inc Header6.h  
#inc Header7.h  
#inc Header8.h  
#inc Header9.h
```

Source1.cxx:

```
#inc Header.h
```

Source2.cxx:

```
#inc Header.h
```

Source3.cxx:

```
#inc Header.h
```

Source4.cxx:

```
#inc Header.h
```

A lot less to compile!

Header Statistics – What's the Unit?

Number #inc: **>100 lines**

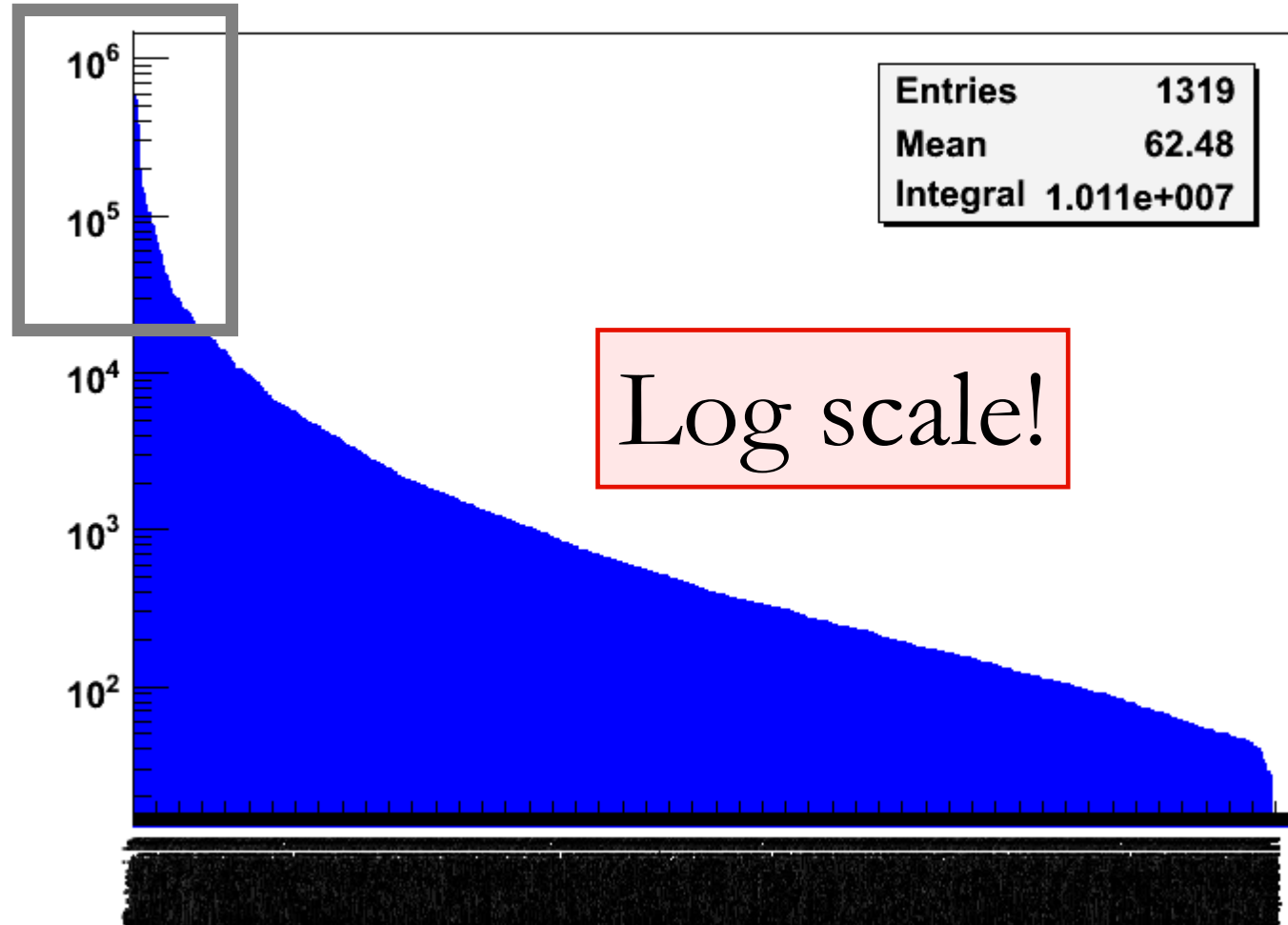
955 RConfig.h
939 RVersion.h
935 DllImport.h
934 Rtypes.h
934 Rtypeinfo.h
915 TGenericClassInfo.h
890 Varargs.h
888 Riosfwd.h
882 TStorage.h
882 TObject.h

Number #inc * lines:

758254 G__ci.h
580957 TMath.h
573705 TString.h
541548 TBuffer.h
534060 Bytes.h
448850 RConfig.h
378270 Rtypes.h
206856 TClass.h
193011 TROOT.h
185220 TObject.h

ROOT Header Statistics

Number of lines
* number of times included



Headers

Ideal Situation

- All headers compiled as included: 10M lines
- All headers compiled once: 0.27M lines = 3%!



All headers as included

once

Context Requirement

ALL compilers need identical context: compiler flags, #included files before

Precompiled

```
Header.h:  
#inc Header1.h  
#inc Header2.h  
#inc Header3.h  
#inc Header4.h  
#inc Header5.h  
#inc Header6.h  
#inc Header7.h  
#inc Header8.h  
  
Source1.cxx:  
  
#inc Header.h
```

Precompiled won't work!

```
Header.h:  
#inc Header1.h  
#inc Header2.h  
#inc Header3.h  
#inc Header4.h  
#inc Header5.h  
#inc Header6.h  
#inc Header7.h  
#inc Header8.h  
  
Source2.cxx:  
  
#inc Header6.h  
#inc Header.h
```

OUCH! →

Context Requirement - Solution

Consequence:

always `#include` the same set of headers for all sources! (whether they need them or not)

Easiest via compiler flag

`-include ForcedHeader.h`

Problem:

Symbol / CPP macro clashes due to new include files

Where's the Optimum?

Somewhere between precompile all / no headers.

Extensive study of include dependencies, compile time, gain. Amazingly simple result:

Precompile only TH1.h and all of its #includes,
= 49 ROOT header files
= 156 headers incl. GCC system headers

Compilers & PCH: GCC/ICC

Compiler dependent, of course. GCC/ICC approach:

1. explicit header precompile step, for GCC:
g++ -c pch.hxx -o pch.hxx.gch
2. output in -I path will be found & used by subsequent compiler calls, i.e. `#include "pch.h"` will allow GCC to use `pch.h.gch`

Flag for forced include: **-include forced.h**

Compilers & PCH: MSVC

1. Create PCH storing everything up to a `#pragma hdrstop`:

```
cl -c src.cxx -Yc -FpROOT.pch
```

2. Use it:

```
cl -c src.cxx -Yu -FpROOT.pch
```

Flag for forced include: **-FI forced.h**

Results

Assume a very fundamental header changes:

touch Rtypes.h && make

Building ROOT/debug without PCH is

- 45% slower: 16 vs. 11min, GCC 4.0.2
- 55% slower: 34 vs. 22min, Win MSVC8

Of course rebuilds due to just one changed source benefit, too.

Summary

So now you know

- what precompiled headers are: compiler's memory
- how to implement them: a few compiler flags, one more file to compile
- and how much they pay off: you loose 50% without them, in realistic cases