Google base 分成许多个子模块

build是编译时模块, build\_config.h 检测编译平台，编译器和处理器架构, 是windows系统会定义OS\_WIN, 编译器是visual\_studio 会定义COMPILER\_MSVC，如果处理器是x64架构会定义\_M\_X64, ARCH\_CPU\_X86\_FAMILY, ARCH\_CPU\_X86\_64, ARCH\_CPU\_64\_BITS, ARCH\_CPU\_LITTLE\_ENDIAN以及宽字符16位的宏WCHAR\_T\_IS\_UTF16.

1.base 基础，公共的头文件和类

macro.h 主要定义了DISALLOW\_COPY DISALLOW\_ASSIGN两个宏

base\_export.h 控制代码的导入导出，Win32 dll接口导出需要预处理定义 COMPOENT\_BUILD, WIN32, BASE\_IMPLEMENTATION. 之后BASE\_EXPORT为导出宏

compiler\_specific.h (build/build\_config.h) 编译器相关的宏

template\_utils.h (build/build\_config.h) 模板辅助类，is\_trivially\_copyable, is\_iterator, SupportsOstreamOperator

callback\_forward.h Callback系列类的前置声明

logging.h (macros.h, callback\_forward.h, strings/string\_piece\_forward.h, debug/debugger.h)日志类，这个类是这个代码库的日志信息基础，基本上所有的源文件都包含了这个类产生调试，日志信息

atomicops.h 原子操作 atomicops\_internals\_x86\_msvc.h

at\_exit.h (callback.h, synchronization/lock.h, bind.h) 程序结束时的回调

bit\_cast.h bit级别赋值

cpu.h CPU信息

location.h (string\_number\_conversion.h)描述一个对象在哪里创建

callback.h (callback\_forward.h, callback\_internal.h) 回调

callback\_internal.h (memory/ref\_counted.h)

atomic\_ref\_count.h 原子级引用计数

base\_switches.h 控制开关

sequence\_token.h 有序任务的标识token

atomic\_sequence\_num.h 线程安全的递增数字生成器

command\_line.h (stringpiece.h, utf\_string\_conversions.h, string\_util.h, string\_split.h, string\_tokenizer.h, files/file\_path.h) 命令行相关

pickle.h (strings/string\_piece.h, memory/ref\_counted.h, bits.h) Pickle Object

bits.h 比特位

vlog.h 包含Logging配置的辅助类

hash.h 哈希算法

single\_thread\_task\_runner.h (sequenced\_task\_runner.h)

sequenced\_task\_runner.h (callback.h, sequenced\_task\_runner\_helpers.h, task\_runner.h)

task\_runner.h (location.h, memory/ref\_counted.h, time/time.h, threading/post\_task\_and\_reply\_impl.h)

run\_loop.h (memory/weak\_ptr.h, observer\_list.h)

observer\_list.h (memory/weak\_ptr.h, stl\_util.h)

callback\_helpers.h (memory/ptr\_util.h)

sys\_info.h (sys\_info\_internal.h, lazy\_instance.h, metrics/field\_trial.h)

version.h

value\_iterators.h

unguessable\_token.h

format\_macros.h

rand\_util.h

environment.h

lazy\_instance.h

bind.h

bind\_internal.h

bind\_helper.h

pending\_task.h

sha1.h

2.string 字符，字符串

string16.h typedef char16 string16

string\_piece\_forward.h (strings/string16.h) StringPiece前置声明

string\_piece.h (string16.h) BasicStringPiece类

stringprintf.h (utf\_string\_conversions.h, string\_util.h, scoped\_clear\_errno.h)

utf\_string\_conversion.h (utf\_string\_conversion\_utils.h, string\_util.h)

utf\_string\_conversion\_utils.h (string16.h, third\_party/icu/icu\_utf.h)

string\_util.h (string16.h, string\_piece.h, memory/singleton.h)

string\_number\_conversions.h (numerics/safe\_math.h, scoped\_clear\_errno.h, third\_party/dmg\_fp/dmg\_fp.h) 数字和字符串的转化

string\_tokenizer.h

string\_split.h (string\_util.h, icu/icu\_utf.h) 字符串分割

sys\_string\_conversions.h 系统宽窄字符相互转换的API

3.debug 调试

debugger.h (threading/platform\_thread.h)几个接口还不是很理解

alias.h

stack\_tracer.h

leak\_annotations.h

activity\_tracker.h (metrics/persistent\_memory\_allocator.h, process/process\_handle)

dump\_without\_crashing.h

task\_annotator.h

4.threading

platform\_thread.h (macros.h, build\_config.h, time/time.h)平台线程相关, PlatformThread

thread\_restrictions.h 线程临界区

thread\_local.h (thread\_local\_storage.h)

thread\_local\_storage.h (synchronization/lock.h)

thread\_collision\_warner.h (atomicops.h) 线程安全的一个Helper Class

thread\_id\_name\_manager.h (synchronization/lock.h, threading/platform\_thread.h, trace\_event/heap\_profiler\_allocation\_context\_tracker.h)

thread\_checker.h thread\_checker\_impl.h (thread\_task\_runner\_handle.h)

thread\_task\_runner\_handle.h (memory/ref\_counted.h, single\_thread\_task\_runner.h)

post\_task\_and\_reply\_impl.h (memory/ref\_counted.h, debug/leak\_annotation.h, sequence\_checker.h, sequenced\_task\_runner.h, threading/sequenced\_task\_runner\_handl.h)

sequenced\_task\_runner\_handle.h (sequenced\_task\_runner.h, lazy\_instance.h, sequenced\_worker\_pool.h, thread\_local.h)

sequenced\_worker\_pool.h (task\_runner.h, task\_scheduler/task\_traits.h)

scoped\_blocking\_call.h

simple\_thread.h (containers/queue.h, synchronization/waitable\_event.h)

5.time 时间

time.h (compiler\_specific.h, build\_config.h, numerics/safe\_math.h) TimeDelta, TimeBase, Time, TimeTicks, ThreadTicks，实现上依赖 strings/stringprintf.h third\_party/nspr/prtime.h lazy\_instance.h

6.numerics 数字，全部头文件

safe\_math.h (checked\_math.h, clamped\_math.h, safe\_conversions.h)

checked\_math.h

clamped\_math.h

safe\_conversions.h

math\_constants.h 两个常量kPiDouble kPiFloat

range.h

7.third\_party

a.nspr. NSPR data and time functions

b.icu

c.dmg\_fp

8.memory 内存管理

singleton.h (macros.h, at\_exit.h, atomicops.h, threading/thread\_restrictions.h)

ref\_counted.h (atomic\_ref\_count.h, sequence\_checker.h, threading/thread\_collision\_warner.h) 引用计数

weak\_ptr.h

ptr\_util.h

aligned\_memory.h

manual\_constructor.h

shared\_memory.h

shared\_memory\_handle.h

shared\_memory\_helper.h

raw\_scoped\_refptr\_mismatch\_checker.h

shared\_memory\_tracker.h

ref\_counted\_memory.h

9.synchronization 同步

lock.h, lock\_impl.h, lock\_impl\_win.cc (debug/activity\_tracker.h)

condition\_variable.h (threading/scoped\_blocking\_call.h, threading/thread\_restrictions.h)

waitable\_event.h (win/scoped\_handle.h, threading/scoped\_blocking\_call.h, activity\_tracker.h)

10.files 文件

file\_path.h (strings/string\_piece.h, string\_util.h, utf\_string\_conversions.h, sys\_string\_conversions.h, pickle.h)文件路径

memory\_mapped\_file.h (file.h, sys\_info.h)

file.h (file\_tracing.h, scoped\_file.h, platform\_file.h, timer/elapsed\_timer.h, metrics/histogram.h)

file\_tracing.h

scoped\_file.h (scoped\_generic.h)

important\_file\_writer.h

11.trace\_event 追踪

heap\_profiler\_allocation\_context\_tracker.h (heap\_profiler\_allocation\_context.h)

memory\_allocator\_dump\_guid.h

memory\_dump\_provider.h

memory\_dump\_provider\_info.h

memory\_dump\_request\_args.h

trace\_config.h

trace\_config\_category\_filter.h

trace\_event.h trace\_event\_impl.h

trace\_catogory.h

trace\_event\_system\_stats\_monitor.h

trace\_log.h

trace\_buffer.h

trace\_event\_argument.h

trace\_event\_memory\_overhead.h

trace\_event\_etw\_export\_win.h

heap\_profiler\_event\_filter.h

trace\_event\_filter.h

memory\_usage\_estimator.h

12.task\_scheduler

task\_traits.h (task\_scheduler/task\_traits\_details.h)

13.containers

queue.h

circular\_deque.h

vector\_buffer.h

flat\_map.h

flat\_tree.h

14.win

scoped\_handle.h

windows\_version.h (registry.h)

registry.h

object\_watcher.h

startup\_information.h

scoped\_process\_information.h

message\_window.h

current\_module.h

wrapped\_window\_proc.h

15.process

process\_handle.h (files/file\_path.h, win/scoped\_handle, win/windows\_version.h)

process\_metrics.h

memory.h

process.h

kill.h

process\_iterator.h

launch.h

process\_info.h

16.json

json\_writer.h

json\_reader.h

json\_parser.h (memory/manual\_constructor.h)

string\_escape.h

json\_string\_value\_serializer.h

17.metrics

persistent\_memory\_allocator.h (memory/shared\_memory.h)

field\_trial.h

histogram.h

histogram\_base.h

histogram\_samples.h

sparse\_histogram.h

statistics\_recorder.h

record\_histogram\_checker.h

histogram\_snapshot\_manager.h

histogram\_flattener.h

metrics\_hashes.h (sys\_byteorder.h, md5.h)

persistent\_histogram\_allocator.h

histogram\_functions.h

18.timer

timer.h

elapsed\_timer.h

19.message\_loop

message\_loop.h

message\_pump.h

timer\_slack.h

message\_loop\_task\_runner.h

incoming\_task\_queue.h