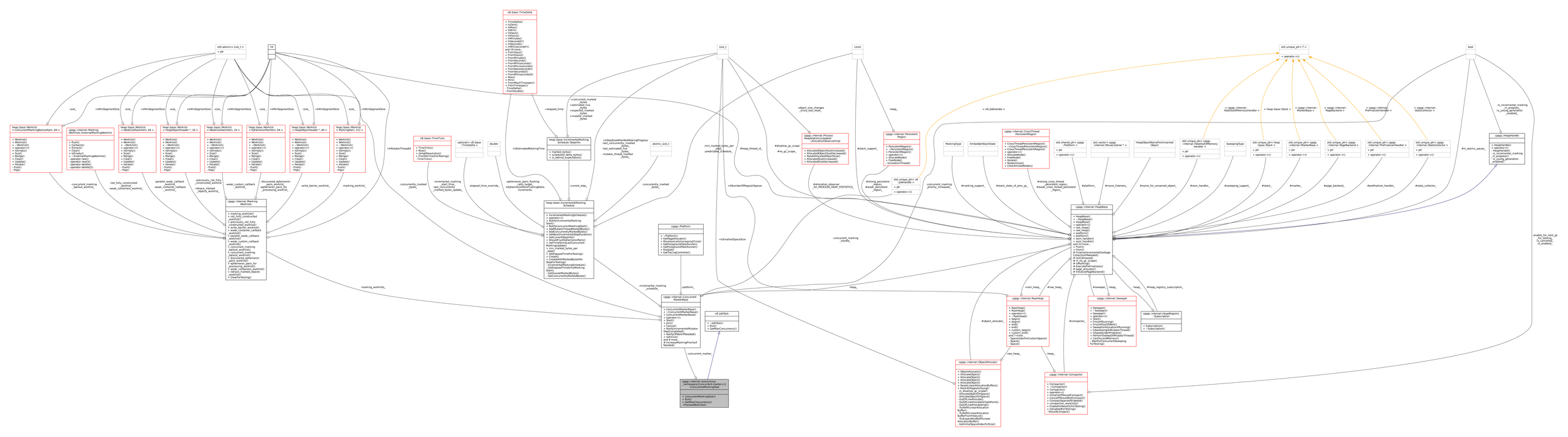
v8::base::TimeDelta
+ TimeDelta()
+ IsZero()
+ IsMax()
+ InMs()
+ InUs()
+ InNs()
+ InHours()
+ InMinutes()
+ InSeconds()
+ InSecondsF()
+ InSecondsFloor()
and 18 more ...
+ FromDays()
+ FromHours()
+ FromMinutes()
+ FromSeconds()
+ FromSecondsD()
+ FromMicroseconds()
+ FromNanoseconds()
+ FromNanosecondsD()
+ FromMillisecondsD()
+ Max()
+ Min()
+ FromMachTicksOnce()
+ FromTicksOnce()
+ FromNanos()
+ FromDouble()
std::atomic< size_t >
+ ptr
int
size_t
const
std::unique_ptr< T >
+ operator->()
bool
-size_
+kMinSegmentSize
heap::base::WorkList
< ConcurrentMarkingBailoutItem, 64 >
+ Push()
+ Contains()
+ IsEmpty()
+ Clear()
+ SizeEstimate()
+ Merge()
+ Swap()
+ Clear()
+ Pop()
+ Iterate()
+ Delete()
- Push()
- Pop()
cppgc::internal::Marking
WorkLists::ExtensivelyMarkingWorkList
+ Push()
+ Contains()
+ IsEmpty()
+ Clear()
+ SizeEstimate()
+ Merge()
+ operator new()
+ operator new[]()
- operator delete()
- operator delete[]()
heap::base::WorkList
< WeakCallback Item, 64 >
heap::base::WorkList
< HeapObjectHeader *, 16 >
heap::base::WorkList
< WeakCallbackItem, 16 >
heap::base::WorkList
< EphemeronPairItem, 64 >
heap::base::WorkList
< HeapObjectHeader *, 64 >
heap::base::WorkList
< MarkingItem, 112 >
-concurrent_marking
_bailout_worklist_
-not_fully_constructed
worklist
-weak_containers_worklist_
-parallel_weak_callback
worklist
-weak_custom_callback
worklist
-previously_not_fully
-constructed_worklist
-retrace_marked
_objects_worklist_
-weak_callback
worklist
-discovered_ephemeron
-pairs_worklist_
-ephemeron_pairs_for
_processing_worklist_
-write_barrier_worklist_
-marking_worklist_
cppgc::internal::Marking
WorkLists
+ marking_worklist()
+ Clear(bool not_fully_constructed
_worklist)
+ previously_not_fully
_constructed_worklist()
+ write_barrier_worklist()
+ weak_container_callback
_worklist()
+ parallel_weak_callback
_worklist()
+ weak_custom_callback
_worklist()
+ DisableMarkingItem
_bailout_worklist()
+ DisableMarkingItem
_bailout_worklist()
+ discovered_ephemeron
_pairs_worklist()
+ ephemeron_pairs_for
_processing_worklist()
+ DisableEphemeronPairs
_worklist()
+ retrace_marked_objects
_worklist()
+ ClearForTesting()
-marking_worklists_
v8::base::TimeTicks
+ Now()
+ HighResolutionNow()
+ IsHighResolution()
+ IsTicksForTesting()
+ TimeTicks()
-concurrently_marked
bytes
-incremental_marking
_last_concurrently
_marked_bytes_update_
+kMutatorThreadA
optional< v8::base
::TimeDelta >
double
-elapsed_time_overide_
+kEstimatedMarkingTime
+elapsed_time
+concurrent_marked
_bytes
+estimated_live
_bytes
+expected_marked
_bytes
+mutator_marked
_bytes
heap::base::IncrementalMarking
Schedule::StepInfo
+ marked_bytes()
+ schedule(_delta_bytes)
+ is_last(no_longer)(also)
-ephemeron_pairs_flushing
-is_forced_finalization
-bytes_markedbeforeincrmeta
increments
+kStepSizeIfNotMakingProgress
-last_concurrently_marked
_bytes
-last_estimated_live
_bytes
-mutator_thread_marked
_bytes
atomic_size_t
-concurrently_marked
bytes
-current_step_
heap::base::IncrementalMarking
Schedule
+ IncrementalMarkingSchedule()
+ ~IncrementalMarking()
+ NotifyIncrementalMarking
Start()
+ AddMultiThreadMarkingStart()
+ AddMultiThreadMarkingBudget()
+ AddConcurrentlyMarkedBytes()
+ GetOverallMarkedBytes(bool)
+ GetCurrentStepSize()
+ ShouldFlushEphemeronPairs()
+ GetIncrementalMarkingTaskRunner()
+ GetBackgroundTaskRunner()
+ Pause()
+ GetTracingController()
-min_marked_bytes_per
-step_size
-predictable_schedule_
+kNumberOfRegularSpaces
+kSmallestSpaceSize
cppgc::internal::Process
HeapStatisticsUpdater
::AllocationObserverImpl
+ AllocatedObjectSizeIncreased()
+ AllocatedObjectSizeDecreased()
+ ResetAllocatedObjectSize()
+ AllocatedBytesIncreased()
+ AllocatedSizeDecreased()
#allocation_observer
_FOR_PROCESS_HEAP_STATISTICS_
-object_size_changes
_since_last_reset_
-heap_
#heap_thread_id_
#disallow_gc_scope
#no_gc_scope_
cppgc::internal::Persistent
Region
+ PersistentRegion()
+ ~PersistentRegion()
+ PersistentRegion()
+ operator=()
+ ClearAllUsed()
+ Freelnode()
+ IsCreationThread()
#stack_support_
#strong_persistent
#weak_persistent
region
std::unique_ptr< v8
::PageHandle >
+ ptr
+ operator->()
< v8::JobHandle >
< cppgc::internal:
FatalOutOfMemoryHandler >
< heap::base::Stack >
< cppgc::internal:
MarkerBase >
< cppgc::internal:
PageBackend >
< cppgc::internal:
PreFinalizerHandler >
< cppgc::internal:
StatsCollector >
-is_incremental_marking
_is_prompt
-is_young_generation
_enabled
cppgc::HeapHandle
+ HeapHandle()
+ ~HeapHandle()
+ HeapHandle()
+ is_incremental_marking
_in_progress()
+ is_young_generation
_enabled()
#in_atomic_pause_
MarkingType
EmbedderStackState
cppgc::internal::CrossThread
PersistentRegion
+ ~CrossThreadPersistentRegion()
+ ~CrossThreadPersistentRegion()
+ operator=()
+ AllocateNode()
+ FreeNode()
+ Iterate()
+ NodesInUse()
+ CheckIsUseBytes()
std::shared_ptr< cppgc
::Platform >
+ ptr
+ operator->()
std::vector< cppgc
::internal::MovableCleaner * >
HeapObjectNameForUnnamed
Object
std::unique_ptr< cppgc
::internal::StatsCollectorMemory
Handler >
+ ptr
+ operator->()
SweepingType
std::unique_ptr< heap
::base::Stack >
+ ptr
+ operator->()
std::unique_ptr< cppgc
::internal::MarkerBase >
+ ptr
+ operator->()
std::unique_ptr< cppgc
::internal::PageBackend >
+ ptr
+ operator->()
std::unique_ptr< cppgc
::internal::PreFinalizerHandler >
+ ptr
+ operator->()
std::unique_ptr< cppgc
::internal::StatsCollector >
+ ptr
+ operator->()
-concurrent_marking
_priority_increaser_
#marking_support_
#stack_state_of_prev_gc_
#strong_cross_thread
#weak_cross_thread_persistent
region
#platform_
#move_listeners_
#name_for_unnamed_object_
#oom_handler_
#sweeping_support_
#stack_
#marker_
#page_backend_
#prefinalizer_handler_
#stats_collector_
-enable_for_next_gc
_for_testing
-is_canceled
-is_enabled_
cppgc::internal::HeapBase
+ ~HeapBase()
+ ~HeapBase()
+ HeapBase()
+ raw_heap()
+ raw_heap()
+ raw_heap()
+ stack()
+ platform()
+ oom_handler()
+ marker() const
and 57 more ...
+ Finish()
+ FinishIncrementalGarbage
CollectionPossible()
# IsGCAllowed()
# IsGCForbidden()
# IsMarking()
# ExecutePreFinalizers()
# AllocateObjectSize()
# InitializeHeapBaseStandard()
-concurrent_marking
handle
-main_heap_
-raw_heap_
#object_allocator_
#compactor_
#sweeper_
-heap_
-platform_
-incremental_marking
schedule
v8::base::
::internal::
MemoryOrder
::kRelaxed
cppgc::internal::Concurrent
MarkerBase
+ ConcurrentMarkerBase()
+ ~ConcurrentMarkerBase()
+ ConcurrentMarkerBase()
+ operator=()
+ Start()
+ Join()
+ Cancel()
+ NotifyIncrementalMutator
StepCompleted()
+ NotifyOfWorkAvailable()
+ JoinForTest()
and 8 more ...
# IncreaseMarkingPriorityIf
Needed()
v8::JobTask
+ ~JobTask()
+ Run()
+ GetMaxConcurrency()
-concurrent_marker_
cppgc::internal::anonymous
_namespace0::(concurrent-marker.cc)
::ConcurrentMarkingTask
+ ~ConcurrentMarkingTask()
+ Run()
+ GetMaxConcurrency()
- ProcessWorkUntil()
cppgc::internal::RawHeap
+ RawHeap()
+ ~RawHeap()
+ operator=()
+ ~RawHeap()
+ begin()
+ begin()
+ end()
+ end()
+ custom_begin()
+ custom_end()
+ Size()
- SpaceIndexForCustomSpace()
- Space()
- Space()
cppgc::internal::Sweeper
+ Sweeper()
+ ~Sweeper()
+ Sweeper()
+ operator=()
+ Start()
+ FinishIfRunning()
+ FinishIfOutOfWork()
+ NotifyDoneIfNeeded(Turning)
+ IsSweepingOnMutatorThread()
+ IsSweepingInProgress()
+ WaitForTasksIfRunning(Thread)
+ CancelIncrementalMemory()
+ WaitForConcurrentSweeping
ForTesting()
cppgc::internal::HeapRegistry
::Subscription
+ Subscription()
+ ~Subscription()
cppgc::internal::Object
Allocator
+ ObjectAllocator()
+ AllocateObject()
+ AllocateObject()
+ AllocateObject()
+ AllocateObject()
+ ResetLinearAllocationBuffers()
+ ~ObjectAllocator()
+ is_disabline_gc_scope()
- AllocateObjectOnSpace()
- OutOfLineAllocate(Block)
- OutOfLineAllocate()
- OutOfLineAllocateGCSafePoint()
- AllocateFromFreelist()
- TryRefillLinearAllocation
Buffer()
- TryRefillLinearAllocation
BufferFromFreeList()
- TryExpandAndRefillLinear
AllocationBuffer()
- GetInitialSpaceIndexForSize()
cppgc::internal::Compactor
+ Compactor()
+ ~Compactor()
+ Compactor()
+ operator=()
+ CancelIfShouldNotCompact()
+ CancelIfShouldNotCompact()
+ CompactSpacesIfEnabled()
+ InitializeIfShouldCompact()
+ EnableforNextGCforTesting()
+ IsEnabledForTesting()
+ UnmoveCompact()