



- abort_on_uncaught_exception_callback
- allow_atomics_wait
- capture_stack_trace_for_uncaught_exceptions
- deoptimizer_lazy_throw
- detailed_source_positions_for_profiling
- enable_ro_allocation_for_snapshot
- flush_denormals
- force_slow_path
- has_fatal_error
- initialized_
- ...

- isolate_addresses
- last_recorder_context_id_

v8::internal::Isolate
<ul style="list-style-type: none">- is_shared_space_isolate_- serializer_enabled_- initialized_from_snapshot_- is_short_builtin_calls_enabled_- owns_shareable_data_- log_object_relocation_- preprocessing_exception_- jitless_- is_frozen_
<ul style="list-style-type: none">+ Isolate()+ operator=()+ operator new()+ operator delete()+ SetUpFromReadOnlyArtifacts()+ set_read_only_heap()+ page_allocator()+ IsCurrent()+ InitializeLoggingAndCounters()+ InitializeCounters()and 412 more...+ InitializeOncePerProcess()+ New()+ New()+ Deinitialize()+ Free()+ Delete()+ CurrentPerIsolateThreadData()+ Current()+ SetCurrent()+ c_entry_fp()and 21 more...- Isolate()- ~Isolate()- Init()- CheckIsolateLayout()- InitializeCodeRanges()- AddCodeMemoryRange()- ForSandbox()- FireCallCompletedCallback

- global_safepoint_next_client_isolate_
- global_safepoint_prev_client_isolate_

#location_



+isolate

+T
+__pad0__

v8::internal::MaybeHandle< T >
<ul style="list-style-type: none">+ MaybeHandle()+ MaybeHandle()+ MaybeHandle()+ MaybeHandle()+ Assert()+ Check()+ ToHandleChecked()+ ToHandle()+ ToHandle()+ equals()+ address()+ is_null()+ MaybeHandle()+ ToHandle()# MaybeHandle()

< v8::internal::Code >

v8::internal::MaybeHandle< v8::internal::Code >
<ul style="list-style-type: none">+ __pad0__+ T+ isolate# location_
<ul style="list-style-type: none">+ MaybeHandle()+ MaybeHandle()+ MaybeHandle()+ MaybeHandle()+ MaybeHandle()+ Assert()+ Check()+ ToHandleChecked()+ ToHandle()+ ToHandle()+ ToHandle()+ equals()+ address()+ is_null()# MaybeHandle()

-is_compiled_

-retain_code_

v8::internal::IsBaselineCompiledScope
<ul style="list-style-type: none">+ IsBaselineCompiledScope()+ IsBaselineCompiledScope()+ is_compiled()