```
Deserializer
# kRootArrayConstantsCount
# kFixedRawDataCount
# kFixedRepeatRootCount
# kHotObjectCount
# kDoubleAlignmentSentinel
# kFirstEncodableFixedRawDataSize
# kLastEncodableFixedRawDataSize
# kFirstEncodableRepeatRoot
# kLastEncodableFixedRepeat
RootCount
# kFirstEncodableVariableRepeat
RootCount
# kEmptyBackingStoreRefSentinel
+ IterateStartupObjectCache()
 IterateSharedHeapObjectCache()
# RestoreExternalReference
Redirector()
# RestoreExternalReference
Redirector()
# CanBeDeferred()
       v8::internal::Serializer
   # sink

    isolate

    hot_objects

    reference_map

    external_reference

    encoder
    root index_map
   code_address_map_code_buffer_

    deferred_objects

    num_back_refs
    dispatch_handle_map_
   and 8 more..
    + Serializer()
    + ~Serializer()
    + Serializer()
    + operator=()
    + Payload()
    + ReferenceMapContains()
    + isolate()
    + cage_base()
    + TotalAllocationSize()
    # IsNotMappedSymbol()
    # SerializeDeferredObjects()
   # SerializeObject()
    # SerializeObjectImpl()
   # MustBeDeferred()
   # VisitRootPointers()
    # SerializeRootObject()
    # PutRoot()
    # PutSmiRoot()
    # PutBackReference()
   and 29 more...
    v8::internal::RootsSerializer
  first_root_to_be_serialized
       _has_been_serialized<sub>_</sub>
  object_cache_index_map
  - can_be_rehashed
  + RootsSerializer()
  + RootsSerializer()
  + operator=()
  + can_be_rehashed()
  + root_has_been_serialized()
  + IsRootAndHasBeenSerialized()
  # CheckRehashability()
  # SerializeInObjectCache()
  # object_cache_empty()
  VisitRootPointers()
  Synchronize()
   v8::internal::StartupSerializer
   shared_heap_serializer
  accessor_infos_function_template_infos
  + StartupSerializer()
      -StartupSerializer()
  + StartupSerializer()
  + operator=()
+ SerializeStrongReferences()
    SerializeWeakReferencesAnd
  Deferred()

    SerializeUsingSharedHeap

  ObjectCache()
  + SerializeUsingStartupObject

    CheckNoDirtyFinalization

  Registries()
  SerializeObjectImpl()
```

v8::internal::Serializer