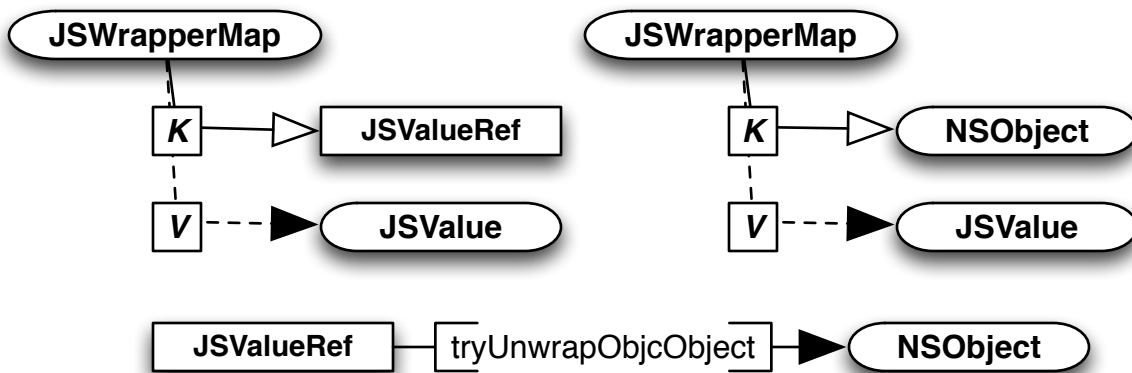


JSWrapperMap

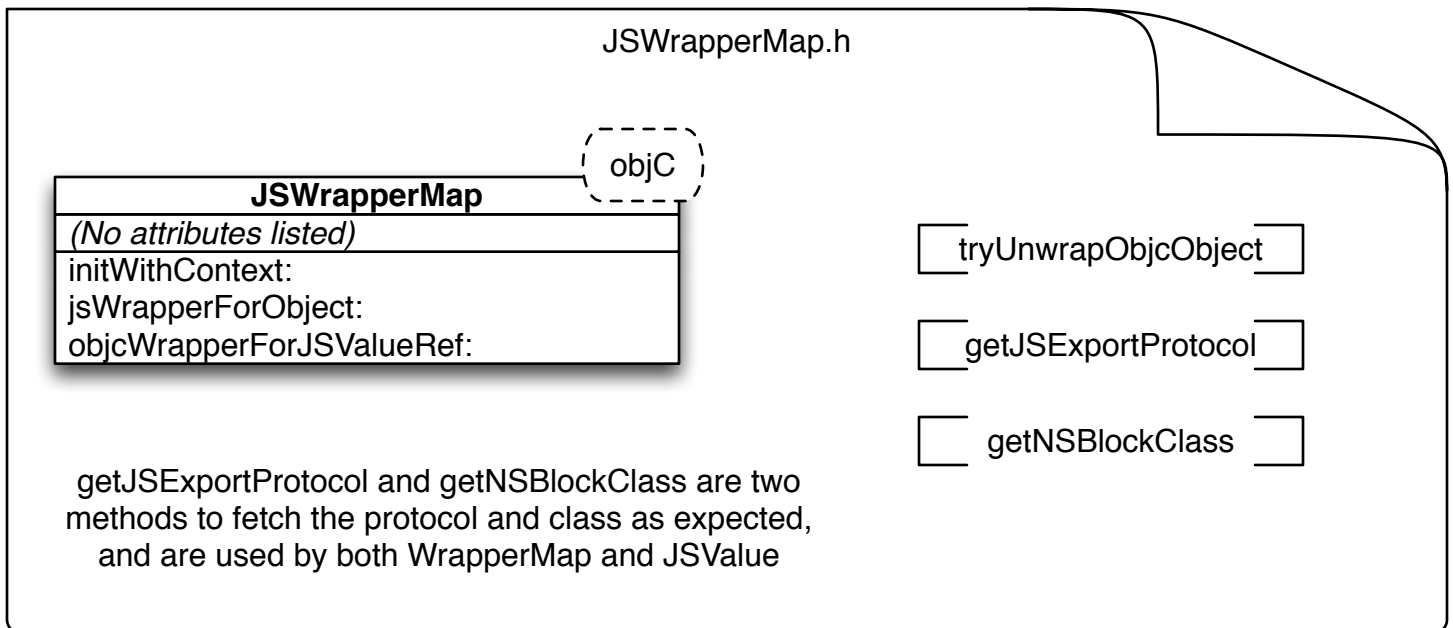
JSWrapperMap is an internal structure used by the JSExport classes to tie native objects to JSValues, tie JSValueRefs to JSValues. Not only does it maintain the relationship, but it also generates new ones if needed, and performs class introspection to generate prototype chains.

Usage

Technically, JSWrapperMap is not public, but the header file exposes fetching a JSValue for an JSValueRef or NSObject. Because such wrappers are generated as needed, it serves as a map that automatically always has a value, and has two key types. A function is also provided to extract a strongly-held NSObject from certain types of Javascript objects.



Diagram

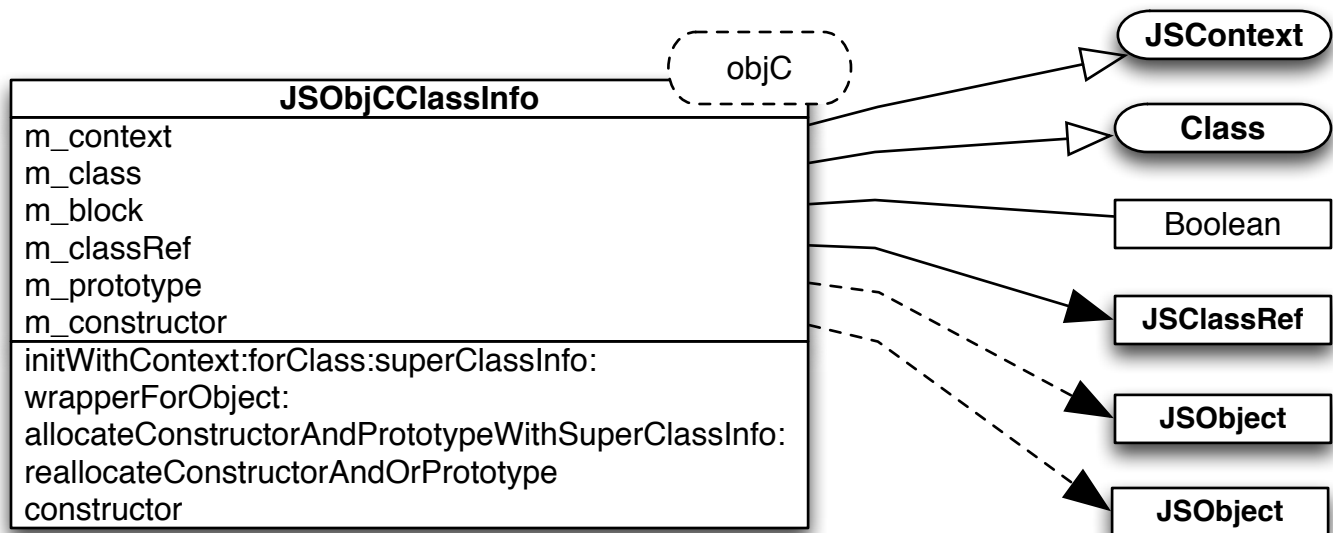


JSWrapperMap

JSWrapperMap.mm

objectWithCustomBrand	makeWrapper	
selectorToPropertyName	makeSetterName	
copyPrototypeProperties	createRenameMap	
copyMethodsToObject	putNonEnumerable	parsePropertyAttributes

These nine functions serve as tools to extract native object, protocol, and class information and to aid in the construction of the JSCore objects, including prototype objects reflecting a native class.



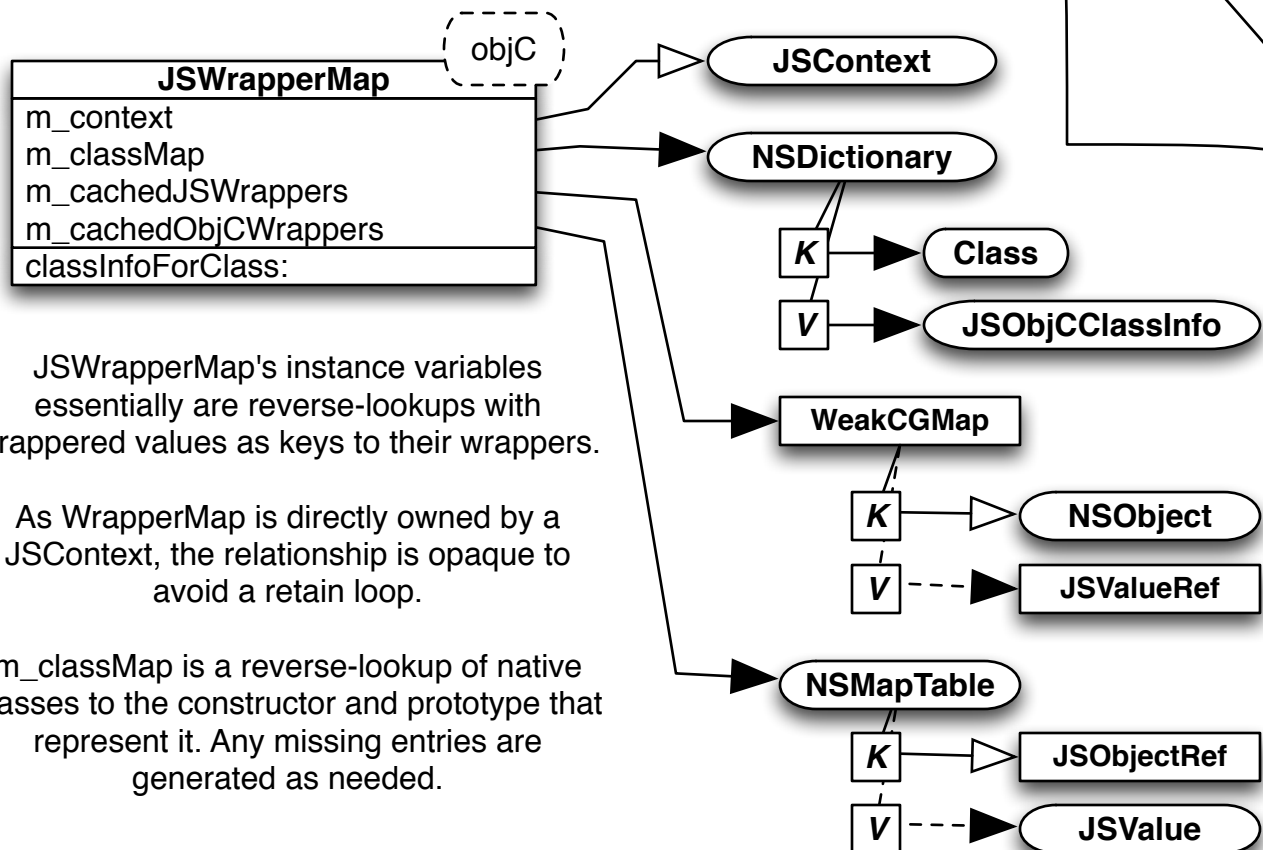
JSObjCClassInfo maintains the relationship between a native class and the JS versions. Because the context indirectly owns this info, we have an opaque relationship to avoid retain loops. Classes are never deallocated so similarly lack a strong relationship. The prototypes and constructors are weakly held, so fetching them causes them to be lazily recreated.

(Continued on page 3)

JSWrapperMap

(Continued from page 2)

JSWrapperMap.mm



JSWrapperMap's instance variables essentially are reverse-lookups with wrapped values as keys to their wrappers.

As WrapperMap is directly owned by a JSContext, the relationship is opaque to avoid a retain loop.

`m_classMap` is a reverse-lookup of native classes to the constructor and prototype that represent it. Any missing entries are generated as needed.

`cachedJSWrappers` is a reverse-lookup of native objects to **JSValueRefs** that represent it. Any missing entries are generated as needed.

`cachedObjCWrappers` is a reverse-lookup of JSObjects to the native **JSValue** that represents it. Any missing entries are generated as needed. Some of these represented **JSObjectRefs** themselves are wrappers for native objects.