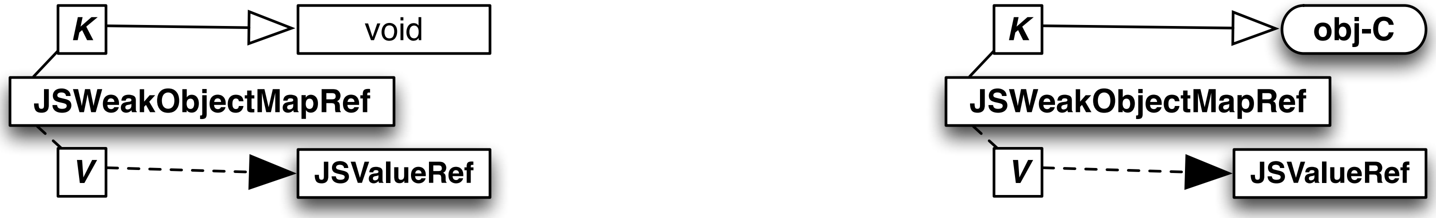
JSWeakObjectMapRef

The object underlying JSWeakObjectMapRef is created via JSWeakObjectMapCreate, and takes two arguements: An opaque data pointer (void *) that is held but not internally used, and a callback that is triggered upon the object's destruction. The callback is handed both the object and the opaque pointer that was passed on creation. The remaining 3 C methods are to manipulate key/value pairs. It uses a "Curiously recurring template pattern" to inherit the reference counting from the RefCounted template, and owns a WeakCGMap for mapping functionality.

Usage

JSWeakObjectMapRef serves as a mapping class between a void* key and a weakly-held JSValueRef. Because of the use of a void* in C++, it's highly likely the intent is to have the key be an opaquely-held Objective C object.



K

void

JSWeakObjectMapRef

V

JSValueRef

K

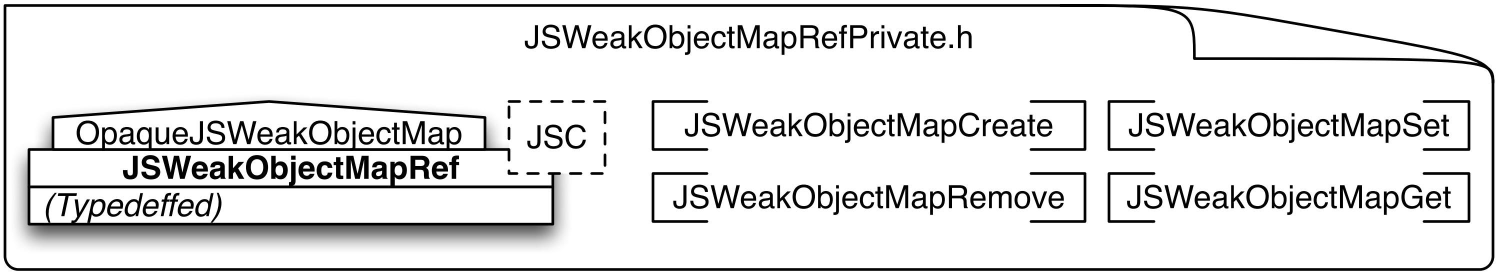
obj-C

JSWeakObjectMapRef

V

JSValueRef

Diagram



JSWeakObjectMapRefPrivate.h

OpaqueJSWeakObjectMap

JSWeakObjectMapRef

(Typedeffed)

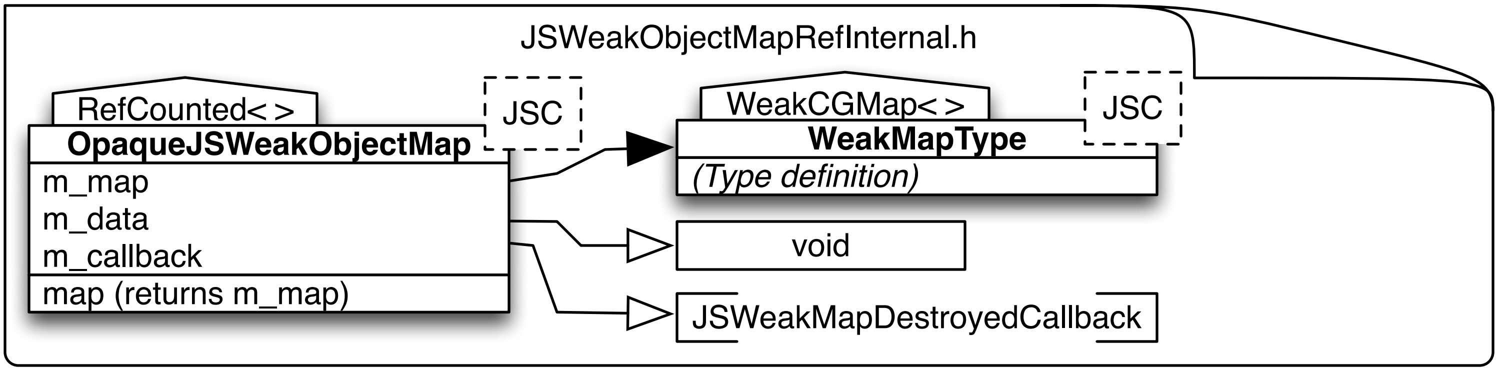
JSC

JSWeakObjectMapCreate

JSWeakObjectMapSet

JSWeakObjectMapRemove

JSWeakObjectMapGet



JSWeakObjectMapRefInternal.h

RefCounted< >

OpaqueJSWeakObjectMap

m_map

m_data

m_callback

map (returns m_map)

JSC

WeakCGMap< >

WeakMapType

(Type definition)

JSC

void

JSWeakMapDestroyedCallback

JSWeakObjectMapRefPrivate.cpp

(No new classes, but implements of functions declared in JSWeakObjectMapRefPrivate.h)