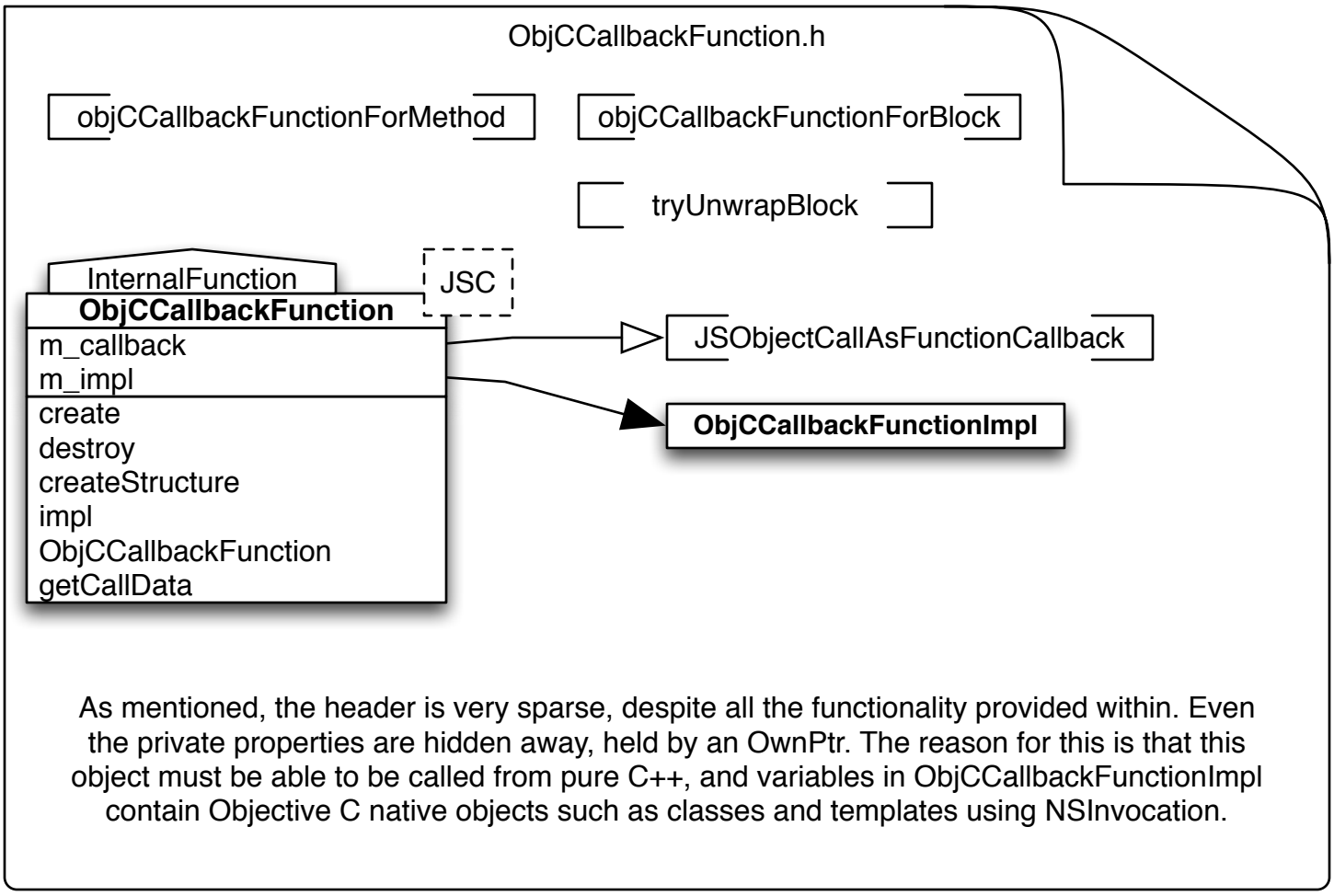
ObjCCallbackFunction

This private JSObjectRef type serves as a wrapper for a native function, either method call or block, and is used by JSWrapperMap. It uses an impressive array of techniques, including polymorphism to construct the argument list within an NSInvocation.

Usage

Very little of ObjCCallbackFunction is provided by the header, save for three methods: one to wrapper a method, another to wrapper a block, and one to attempt to extract the block.

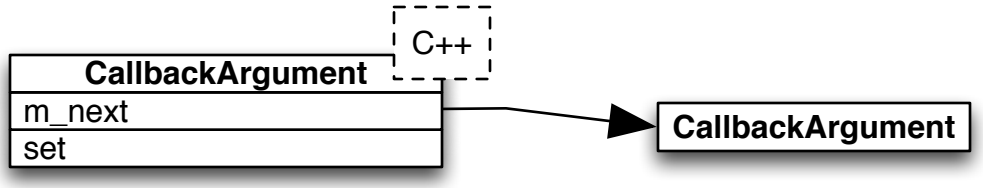
Diagram



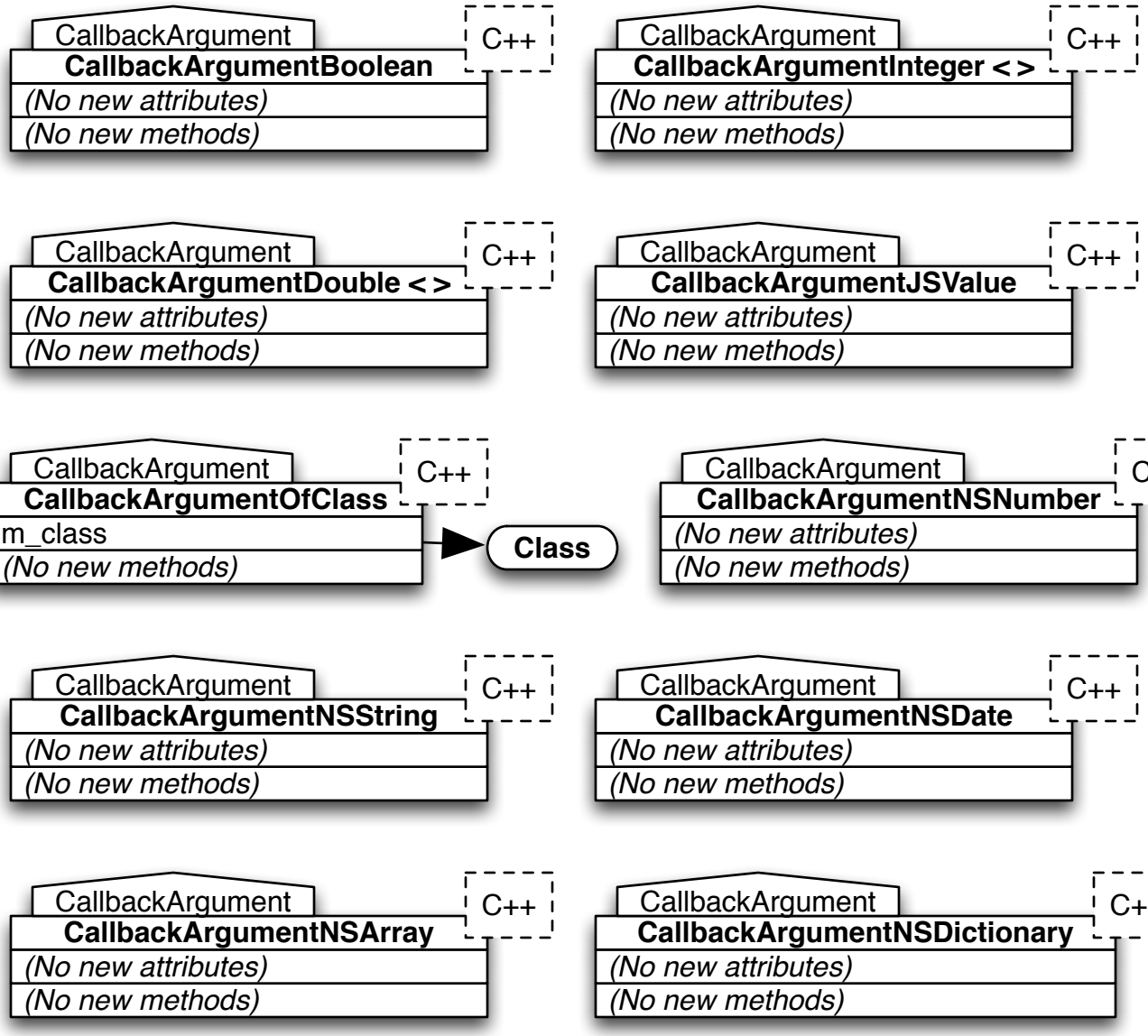
As mentioned, the header is very sparse, despite all the functionality provided within. Even the private properties are hidden away, held by an OwnPtr. The reason for this is that this object must be able to be called from pure C++, and variables in ObjCCallbackFunctionImpl contain Objective C native objects such as classes and templates using NSInvocation.

ObjCCallbackFunction

ObjCCallbackFunction.mm



CallbackArgument is an abstract class and acts as a node in a linked list. The subclasses override set to convert JSValueRef and store the result into an invocation argument

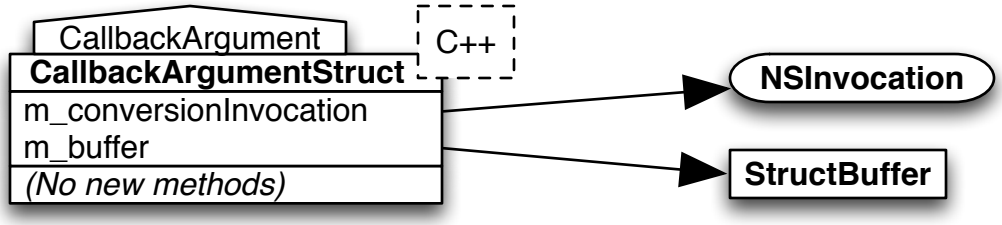


(Continued on page 3)

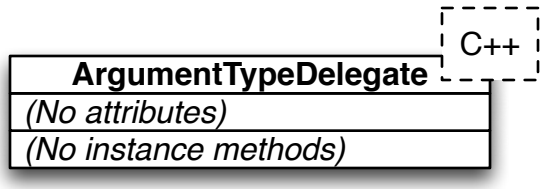
ObjCCallbackFunction

(Continued from page 2)

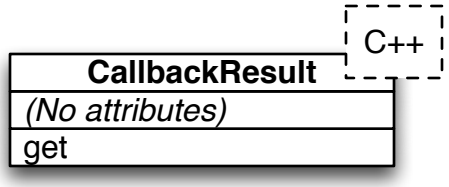
ObjCCallbackFunction.mm



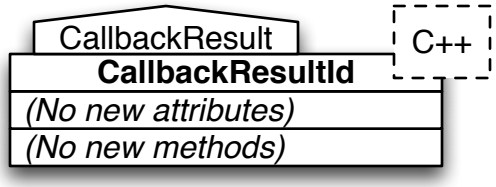
CallbackArgumentStruct stores both the converting function from JSValue and a buffer whose alignment and size are calculated from the datatype.

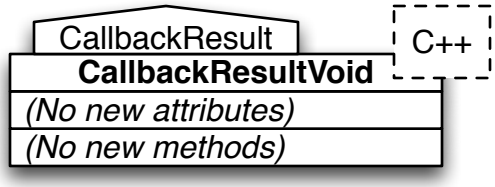


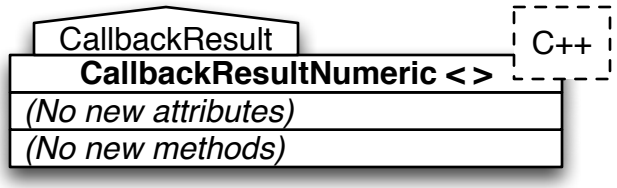
ArgumentTypeDelegate has only static functions, and is used to generate the appropriate CallbackArgument subclass through template magic in ObjcRuntimeExtras.h's parseObjCType function.

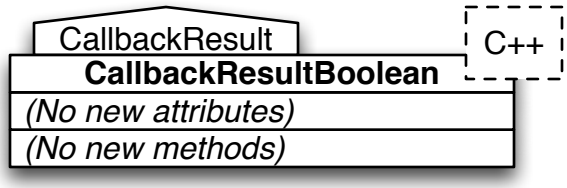


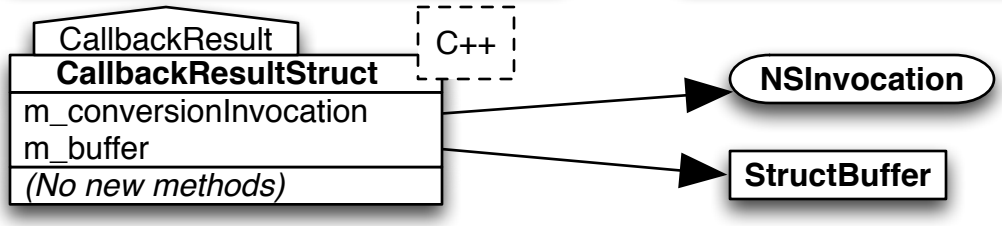
CallbackResult is the mirror class to CallbackArgument. Since there is only one result to a function, this is not a linked list. Furthermore, type checking is not needed. As such, there are different subclasses.









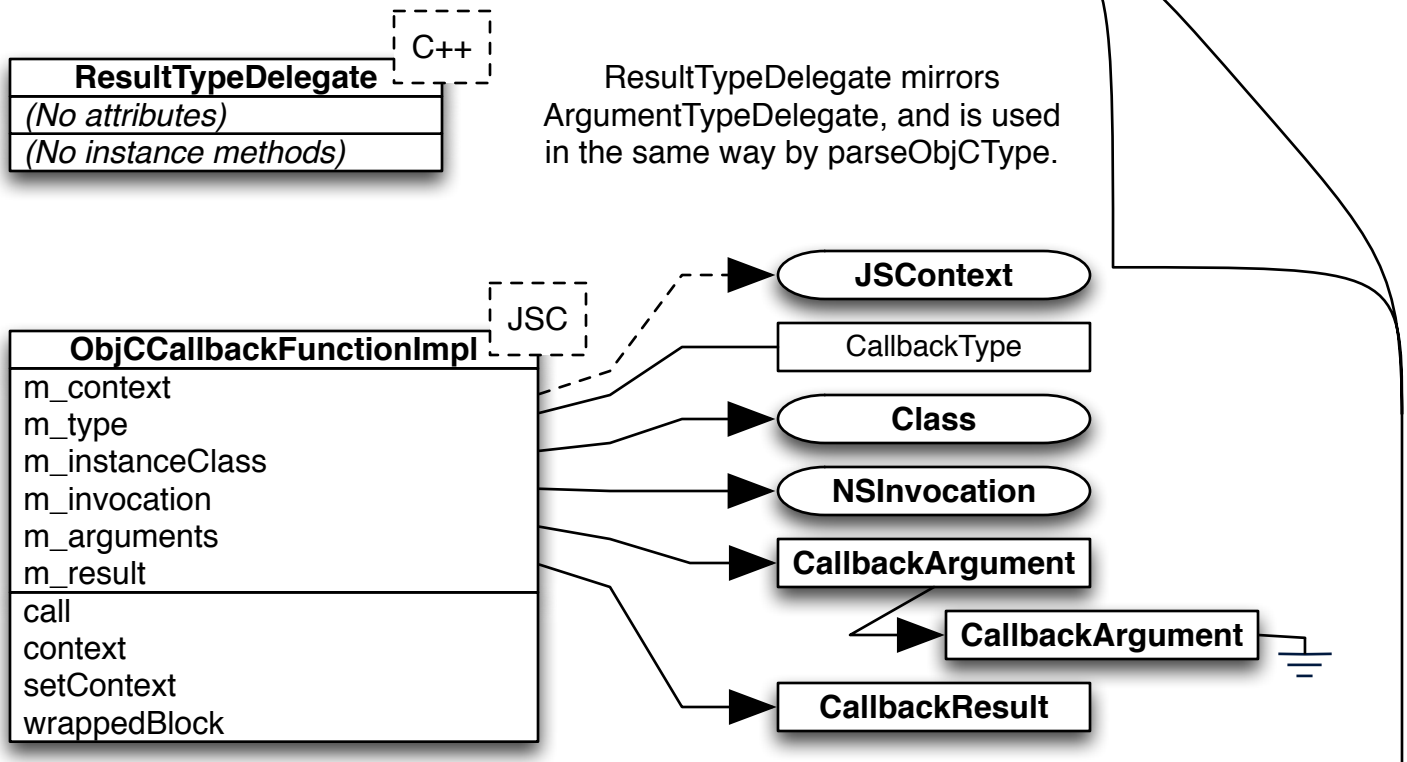


(Continued on page 4)

ObjCCallbackFunction

(Continued from page 3)

JSValue.mm



ObjCCallbackFunctionImpl performs the heart of ObjCCallbackFunction's duties. CallbackType is an enumeration, specifying instance method, class method, or block. The invocation is called via the call method, which runs the argument chain through the callbackArgument linked list, processing the result in callbackResult.

objCCallbackFunctionCallAsFunction

The actual invoking of the function via Javascript is done through objCCallbackFunctionCallAsFunction, which does some sanity checking and preparation.

blockSignatureContainsClass skipNumber

BlockSignatureContainsClass and skipNumber are helper functions used internally by objCCallbackFunctionForInvocation.

objCCallbackFunctionForInvocation

Actual creation of ObjCCallbackFunction is done by objCCallbackFunctionForMethod and objCCallbackFunctionForBlock, which both create NSInvocations before passing the duty onto ObjCCallbackFunctionForInvocation.