# Promises in V8 [OBSOLETE]

Yutaka Hirano <yhirano@chromium.org>

## Objective

I would like to propose implementing Promises in V8.
This document describes the design proposal.

## Background

Promises are being specified at
https://github.com/domenic/promises-unwrapping/blob/master/README.md .
Promises are expected to be parts of ECMAScript6.

## Proposed Design

I propose to implement Promises with JavaScript, in “src/promise.js”.
Promises are implemented behind a flag “harmony_promises”.
Some native functions are provided to the script to implement Promise.

### ES6 Features

Promises use some ES6 features such as iterable objects. They are already implemented, but are marked as experimental. Since Promises are experimental as well and all ES6 features will be shipped simultaneously, we can use experimental features to implement Promises with flag implications.

### ES6 Features [Alternative Design]

Promises use some ES6 features such as iterable objects. They are already implemented, but currently marked as experimental. We implement some native functions to make Promises enable to access such features without flags.

Pros

- harmony_promises is independent of other flags.

Cons

- Implementation gets a bit more complicated and it should be rewritten at some point.

### Thenable Coercions

The spec uses WeakMaps to impliment Thenable Coercions. We use V8 hidden properties to

implement it. Empty entries in WeakMaps will be sweeped only in major GC, so using hidden properties will decrease memory consumption.

## JSPromise

I implement JSPromise, a JSObject subclass in objects.cc.
JSPromise has Promise properties and JS Promise implementation uses runtime functions to access the properties.

## Runtime Functions provided for Promise implementation

### IsPromise(object)

Returns true if |object| is a Promise object.

### IsPromiseInitialized(promise)

Returns true if |promise| is initialized.

### PromiseGetValue(promise)

Asserts that |promise| is fulfilled.
Returns |promise.[[Value]]|

### PromiseGetReason(promise)

Asserts that |promise| is rejected.
Returns |promise.[[Reason]]|

### PromiseGetFollowing(promise)

Asserts that |promise| is following another Promise.
Returns |promise.[[Following]]|

### PromiseHasValue(promise)

Returns true if |promise| is fulfilled.

### PromiseHasReason(promise)

Returns true if |promise| is rejected.

### PromiseIsFollowing(promise)

Returns true if |promise| is following another Promise.

### PromiseGetDerived(promise)

Returns |promise.[[Derived]]|

### PromiseInitialize(promise)

Asserts that |promise| is not initialized.
Initializes |promise| and returns true.

### PromiseSetValue(promise, value)

Asserts that |promise| is not fulfilled nor rejected.
Sets |promise.[[Value]]| to |value|.
Unlike SetValue function in the spec, this function does NOT propergate the value to derived promises.

### PromiseSetReason(promise, reason)

Asserts that |promise| is not fulfilled nor rejected.
Sets |promise.[[Reason]]| to |reason|.
Unlike SetReason function in the spec, this function does NOT propergate the value to derived promises.

### PromiseSetFollowing(promise, following)

Asserts that |promise| is not fulfilled, rejected nor following.
Sets |promise.[[Following]]| to |following|.

### PromiseQueueTask(function)

Queue a microtask to run |function|.

## Proposed API

```
class V8_EXPORT Promise : public Object {
 public:
  // Registers on_fulfilled and on_rejected functions to
  // this Promise so that each of them will be called when
  // this Promise is fulfilled or rejected respectively.
  // Returns a Promise which will be fulfilled or rejected
  // when this Promise is fulfilled or rejected.
  //
  // This method corresponds to Promise.prototype.then in ES6
  // and hence recursively unwraps promises.
```

```
  Local<Promise> Then(Handle<Value> on_fulfilled,
                      Handle<Value> on_rejected);

  // Creates a Promise.
  static Local<Promise> New();

  // Resolves this Promise with |value|.
  // Does nothing if this Promise is already resolved or rejected.
  void Resolve(Handle<Value> value);

  // Rejects this Promise with |reason|.
  // Does nothing if this Promise is already resolved or rejected.
  void Reject(Handle<Value> reason);

  V8_INLINE static Promise* Cast(Value* obj);

 private:
  ...
};
```

# Code Location

- src/promise.js
  - JS script implementing Promise.
- src/objects.cc
  - Define JSPromise.
- src/runtime.cc
  - Runtime functions are implemented here.

# Testing

Many features are tested using cctest.
Some features that can be tested synchronously are tested using mjsunit.

# Performance

To Be Written
http://thanpol.as/javascript/promises-a-performance-hits-you-should-be-aware-of/

# Discussions

## Implementation in C++ vs. JS

Adam is worried that a JS implementation will leak its implementation details.
We agreed to try out prototype implementation in JS and inspect it.

## Revision History

2013-10-15 first draft