N-API On JerryScript

Gabriel Schulhof, Intel of Canada, Ltd.
Node.js Collaborator, TSC Member
What is JerryScript?

A JavaScript VM For Constrained Devices

• Requirements:
  • Small footprint (< 64 K RAM, < 200 K flash)
  • Small size is more important than high speed
  • Fixed heap size

• Features
  • ES 5.1
  • Missing: Symbol, ES6 constants
  • Explicit references - no scopes
What is N-API?

A set of APIs for running native code from Node.js

- **ABI-stable**
  - Compile once, run against current major version of Node.js as well as all future *major* versions

- **VM-agnostic**
  - The underlying JavaScript VM (V8, ChakraCore, JerryScript, SpiderMonkey, etc.) does not matter
JerryScript Requirements for N-API

- Where to put napi_env? → `jerry_get_context_data()`
- Module support → `jerryx_module_resolve()`
- Test conversion → custom Node.js script
- Weak references → `jerry_objects_foreach_by_native_info()`
jerry_get_context_data()

- Implicit context (thread-local)
- Need to "hang" data from it and retrieve it reliably
  → type-tag data with pointer to user data

```c
static jerry_context_data_manager_t
my_data_manager = {
    /*optional*/ init_cb
    /*optional*/ deinit_cb
    /*optional*/ finalize_cb
    /*required*/ sizeof(per_context_data_t)
};
```
Module Support - FEATURE_INIT_FINI

• Registration vs. initialization
  • Node.js:
    • ... → require() → register → initialize → use
  • JerryScript
    • register → ... → require() → initialize → user

• Two-level implementation
  • jerryx_module_resolve(name, resolvers, resolvers_count);
  • 1.) Resolver → 2.) Module
  • Default: jerryx_module_native_resolver and
    jerryx_native_module_t via JERRYX_NATIVE_MODULE(name, init)
    where init is of the form jerry_value_t init (void);
Module Support - Trace-through

- the process starts
- the modules register (`jerryx_native_module_register ()`)
  ...
- JavaScript calls `require(moduleName)`, a native function
- The native function calls `jerryx_module_resolve (name, resolvers, resolvers_count)`
- The native resolver
  - finds the module
  - calls its `init()`
  - returns the resulting `jerry_value_t`
- `jerryx_module_resolve ()` returns the `jerry_value_t`
- `require(moduleName)` returns the `jerry_value_t`
N-API Test Conversion

• Generate CMake files from gyp files (one executable target for each test file)
• Relax compiler flags
• Convert down to ES5.1 (babel)
• C and JavaScript boilerplate
  • global
  • global.gc()
  • setImmediate
  • console
Weak References

- `jerry_acquire_value ()` → strong reference
- `jerry_release_value ()` → weak reference

Given: `jerry_value_t` holding an object
Question: Is it valid?
Answer: `jerry_objects_foreach_by_native_info ()`
Weak References

```
jerry_value_t (213)
info_p@0x330ff301
```

```
  napi_ref {
    .value = 213
    .info_p = 0x330ff301
  };
```

```
  ...  
```

```
jerry_value_t (213)
```

```
  napi_ref {
    .value = 213
    .info_p = 0x330ff301
  };
```

```
  ...  
```

```
jerry_value_t (213)
```

```
  napi_ref {
    .value = 213
    .info_p = 0x330ff301
  };
```

```
info_p@0x450e099b
```

```
  x
```

```
info_p@0x450e099b
```

```
  x
```

```
info_p@0x450e099b
```

```
  x
```

```
```
The Code

https://github.com/gabrielschulhof/jerryscript/tree/n-api-take-2
Shout-out to ShadowNode

https://github.com/Rokid/ShadowNode
Thank You