

Getting Started with N-API and **node-addon-api**

A Hands-on Workshop



About Gabriel Schulhof

- Works at Intel
- Involved with the API working group
 - Promises
 - Exception handling
 - Environment propagation
 - Module loading
 - Wrap/Unwrap
- GitHub: @gabrielschulhof
- Email: gabriel.schulhof@intel.com



About Michael Dawson

IBM Community Lead for Node.js

- Active Node.js community member
 - Technical Steering Committee TSC member
 - Community Committee member
 - n-api, build, security, benchmarking, diagnostics, release, user-feedback, teams and WGs.

Twitter: @mhdawson1

GitHub: @mhdawson

Linkedin: <https://www.linkedin.com/in/michael-dawson-6051282>



About Jim Schlight

inspiredware

- Head of a consultancy based in Ashland, Oregon
- Member of the N-API Working Group
 - node-pre-gyp
 - prebuild
 - Documentation
- GitHub: @jschlight
- Twitter: @inspiredware



About Nicola Del Gobbo

- Developer at Packly
- Member of the N-API Working Group
- GitHub: @NickNaso
- Twitter: @NickNaso
- LinkedIn: <https://it.linkedin.com/in/ndelgobbo>

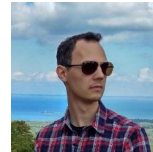
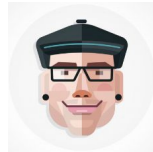
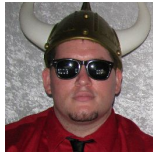
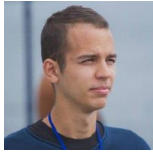
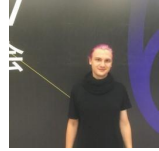


About Hitesh Kanwathirtha

- Software Engineering Manager on the Chakra JavaScript Runtime team
- Involved in Node-ChakraCore and the early N-API efforts
- Github: @digitalinfinity



Contributors



Workshop schedule

- Introduction to N-API and node-addon-api
- Online tutorials
- Let's port some modules/individual projects
- Wrap-up and assessment



Objectives of the workshop

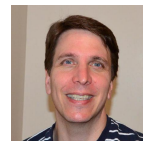
- Orientation to N-API
- Awareness of available tools and processes
- A good start on your own projects



Other Talks to learn More

Building Next Generation Add-on Modules for Node.js - 1:35 PM Day 0 (Monday)

The Next Generation Node API Is Ready – 2:20 PM Day 1 (Tuesday)



Motivations for N-API

- Abstraction of the underlying JavaScript engine
- A binary-stable ABI



Without N-API

Addon

Node.js: 6.12.2
V8: 5.1.281
NODE_MODULE_VERSION: 48



Without N-API

- ABI break
 - At worst, mysterious segfaults
 - At best, addon fails to load
 - `NODE_MODULE_VERSION` mismatch

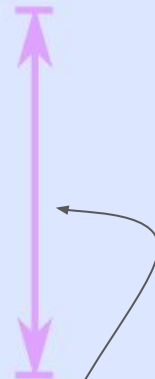
Node.js: 8.9.3

V8: 6.1.534

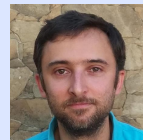
`NODE_MODULE_VERSION`: 57



With N-API



Node.js: no matter
V8/ChakraCore: no matter
NAPI_VERSION: 1



With N-API

- No ABI break
 - `NAPI_VERSION` is cumulative
 - Addon is forwards compatible

Node.js: later
V8/ChakraCore: no matter
N-API_VERSION: 2



A brief history of N-API

- Kicked off in VM Summit April 2016
 - Dual implementation from the start (V8/ChakraCore)
- EPS Dec 2016 - <https://github.com/nodejs/node-eps/blob/master/005-ABI-Stable-Module-API.md>
- First half of 2017 - refinement and backporting modules
- VM Summit July 2017 - Exit criteria agreed
 - <https://github.com/nodejs/node/issues/14532>
- N-API leaves experimental status March 14 2018
 - Stable in master and 10.x
- Backported to 6.x, 8.x
 - 6.x – doc's only experimental
 - 8.x - still needs SemVer minor to be stable in 8.x
- Current focus node-addon-api doc/testing



N-API and node-addon-api

- N-API is C whereas node-addon-api is C++
- node-addon-api not technically part of ABI stable API
 - Helper to simplify code using C++
 - All inline
 - Only depends on export N-API C functions



N-API and node-addon-api

- The code in your addon:

```
1 Object obj = Object::New(env);  
2  
3 obj["foo"] = String::New(env, "bar");
```



N-API and node-addon-api

- What actually gets compiled after inline substitution:

```
1 napi_status status;  
2 napi_value object, string;  
3 status = napi_create_object(env, &object);  
4 if (status != napi_ok) {  
5     napi_throw_error(env, ...);  
6     return;  
7 }  
8 status = napi_create_string_utf8(env, "bar", NAPI_AUTO_LENGTH, &string);  
9 if (status != napi_ok) {  
10     napi_throw_error(env, ...);  
11     return;  
12 }  
13 status = napi_set_named_property(env, object, "foo", string);  
14 if (status != napi_ok) {  
15     napi_throw_error(env, ...);  
16     return;  
17 }
```



Node-addon-api Versus NAN

- Same
 - Includes a C++ wrapper
 - Provides common helper functionality
 - Reduces likelihood of needing to change code for new Node.js versions
- Different
 - Does not use V8 Types
 - Not tied to a specific JavaScript engine
 - Preserves compile once/run multiple versions from N-API
 - Reduces even further likelihood of having to change code



Online Tutorials

- For starting a new add-on module from scratch
 - napi.inspiredware.com/getting-started/first.html
- For migrating an existing NAN module
 - napi.inspiredware.com/getting-started/migration.html
- Async Worker
 - napi.inspiredware.com/special-topics/asyncworker.html
- Node pre-gyp
 - <https://napi.inspiredware.com/special-topics/node-pre-gyp.html>



Opportunity to work on individual projects

- Workshop presenters available for help and questions
- List of candidate modules to port



Supporting resources

- N-API Documentation
 - <https://nodejs.org/api/n-api.html> — The C API
 - <https://github.com/nodejs/node-addon-api> — The C++ wrapper
- N-API migration assistant
 - <https://github.com/nodejs/node-addon-api/blob/master/doc/conversion-tool.md>
- Generator
 - <https://www.npmjs.com/package/generator-napi-module>
- Examples
 - <https://github.com/nodejs/abi-stable-node-addon-examples>
- Node-pre-gyp Documentation
 - <https://github.com/mapbox/node-pre-gyp>
- Prebuildify
 - <https://www.npmjs.com/package/prebuildify>



Wrap-up and assessment

What did you like/not like

- Workshop
- N-API

Get Involved:

- <https://github.com/nodejs/abi-stable-node>
- <https://github.com/nodejs/node>
- <https://github.com/nodejs/node-addon-api>
- Weekly N-API meeting - Thursday 1:30 EST

https://calendar.google.com/calendar/embed?src=nodejs.org_nr77ama8p7d7f9ajrpnu506c98%40group.calendar.google.com

