Building a Node.js Windows C/C++ Addon

http://coderesearchlabs.com/articles/BNWCA.pdf

Code Research Laboratories
www.coderesearchlabs.com

Javier Santo Domingo
j-a-s-d@coderesearchlabs.com

Buenos Aires, Argentina

started - October 15th, 2011
published - October 15th, 2011
last revision - November 19th, 2011

INTRODUCTION

I just finished compiling a node.js addon for Windows. So, since it took me about 4 hours to get this working and to avoid someone else spents all that time too to get a simple "hello world" node.js addon being compiled, I'm writing this simple instructions. I will try to write them for newbies and/or coders don't used to Windows programming as much as possible, placing notes everywhere, showing how I compiled the simple addon source code included at: http://nodejs.org/docs/latest/api/addons.html.

> *NOTE: Besides this steps on how to build an addon on Windows, you might be interested also in my project **NodeJS Addonis** (http://coderesearchlabs.com/nodejsaddonis), which is a simplifier C wrapper for the nodejs addons construction. Check it out!*

STEPS

1. Download Node.JS source code

http://nodejs.org/#download

NOTE: The official distribution supports Windows DLL Addons since version 0.6.0, but if you want to make use of your addons with the 0.5.x series, just download the Node.js fork branch "windowsdll" made by Bradley Meck:

https://github.com/bmeck/node/tree/windowsdll

Be sure you understand that the fork belongs to the node.js version 0.5.7 -unstable-, so that's the node.exe you are going to obtain. To obtain other 0.5.x version supporting DLL you have to manually apply all the changes already made in this fork branch or code your own DLL support (add support for dlopen in node.cc, call LoadLibrary/UnloadLibrary, etc).

2. Build it to obtain the node.lib file. For doing this you will need:

2.1 Download and install (if you don't have them already):

2.1.1 VC++ 2010 Express from:

http://www.microsoft.com/visualstudio/en-us/products/2010-editions/visual-cpp-express

2.1.2 Python 2.x from:

http://www.python.org/download/

NOTE: Be sure it is not 3.x since GYP expects 2.x. In my case I used Python 2.7.2

2.2 Run the vcbuild.bat file

NOTE: Be sure the Python installation path is in the PATH environment variable

2.3 You will get node.lib (amongst other files like node.exe) into Debug or Release directories

3. Create a DLL project in Visual C++

3.1 Go to: File \ New Project

3.2 And then pick: Visual C++ \ Win32

3.3 And after that: Win32 Project

3.4 In Win32 Application Wizard \ Application Settings set Application Type to DLL

NOTE: In my case I called my addon with the original name of "myaddon"

4. Write your addon code

4.1 Add your code. In my case I just adapted the example at the documentation:

```
extern "C" void NODE_EXTERN init (Handle<Object> target)
{
	HandleScope scope;
	target->Set(String::New("hello"), String::New("world"));
}
```

NOTE: Do not forget NODE_EXTERN (which is defined as __declspec(dllexport) in src/node.h) otherwise you will get the error "No module symbol found in module." when trying to load the module via require()

4.2 Add the reference to node.lib to your code. I made it with the line:

```
#pragma comment(lib, "c:\\node.js\\src\\Debug\\node")
```

NOTE: In my case I added it to the file called "myaddon.cpp". This is an important step, otherwise you will get a ton of unresolved external symbols from the linker

VC++ HINT: You can add the Library Directories under the project's Property Pages \ Configuration Properties \ VC++ Directories. In my case node.lib was at: "c:\node.js\src\Debug", so the pragma line can be reduced to a more elegant #pragma comment(lib, "node")

4.3 Add the Include Directories or adjust the headers inclusions to match yours so the compiler can find them

NOTE: In my case I just quickly adjusted all of them (they are just a few) to point directly to their full path. For example when it was including v8.h I replaced it with c:\node.js\src\deps\v8\include\v8.h, when it was including node_object_wrap.h I replaced it with c:\node.js\src\src\node_object_wrap.h, and so on

VC++ HINT: You can add the Include Directories under the project's Property Pages \ Configuration Properties \ VC++ Directories. In my case they are: "c:\node.js\src\deps\uv\include\;c:\node.js\src\deps\v8\include\;C:\node.js\src\src\"

5. Build the addon

5.1 Build it and you will get your addon compiled as a .dll file (at Debug directory if you compiled under the DEBUG profile)

VC++ HINT: Just press F7

5.2 Rename it as .node

VC++ HINT: You can change the Target Extension under the project's Property Pages \ Configuration Properties \ General, so you can get it directly as .node

5.3 Copy it to the same directory of the node.exe generated previously

NOTE: This is to make it simple to test it by requiring from ./

VC++ HINT: You can change the Output Directory under the project's Property Pages \ Configuration Properties \ General, so you can get it directly in the same directory than node.exe

5.4 Run the addon

NOTE: In my case I get the following output:

```
C:\node.js>node
> require("./myaddon").hello
'world'
```