node-rng

A C++ module for node.js to provide access to a high quality hardware random number generator. At this time only the Intel Secure Key with Ivy Bridge hardware is supported.

Quick Usage

```
var rng = require("rng");

if(rng.isAvailable())
{
    var randomNumber    = rng.getRandom();
    randomNumber        = rng.getRandomRange(-10, 10);

    rng.getRandomAsync(function(result) {
        console.log(result);
    });

    rng.getRandomRangeAsync(-10, 10, function(low, high, result) {
        console.log(result);
    });
}
```

API

isAvailable()

Returns `true` if the hardware random number generator is available.

getCorrections()

Returns the number of times the hardware detected a poor quality random number and corrected it before supplying the client. Repeated non-zero values can indicate a problem with the hardware.

getVersion()

Returns the major, minor, and patch version of the module as a SemVer friendly string.

getRandom()

Returns an unsigned 32-bit random number in the interval of [0, 4,294,967,295] synchronously.

getRandomAsync(function(error, result))

Calls `function` with an unsigned 32-bit random number in the interval of [0, 4,294,967,295] asynchronously. The `error` field is `null` if no correction was necessary before supplying a valid random number. Otherwise it will contain an error exception.

getRandomRange(lower, upper)

Returns a signed 32-bit random number in the interval of [‘lower’, ‘upper’] synchronously such that `lower` can be a minimum of -2,147,483,648 and `upper` a maximum of 2,147,483,647.

getRandomRangeAsync(lower, upper, function(error, result))

Returns a signed 32-bit random number to `function` asynchronously in the interval of [‘lower’, ‘upper’] asynchronously such that `lower` can be a minimum of -2,147,483,648 and `upper` a maximum of 2,147,483,647. The `error` field is `null` if no correction was necessary before supplying a valid random number. Otherwise it will contain an error exception.

Building

Building node.js

- Official node.js debianized packages are either poorly packaged or woefully out of date at best on my distro. At this time, must build from source.

- Build node.js from source on a typical GNU system.

```
    $ wget https://nodejs.org/dist/v5.8.0/node-v5.8.0.tar.gz
    $ tar xvzf node-v5.8.0.tar.gz
    $ cd node-v5.8.0
    $ ./configure --debug --prefix=$HOME/.local
    $ make && make check && make install
```

- Add node-gyp to search path in ~/.bashrc, if you prefer

```
    # Add local node.js build...
    if [ -e $HOME/.local/lib/node_modules/npm/bin/node-gyp-bin/node-gyp ] ; then
        PATH=$HOME/.local/lib/node_modules/npm/bin/node-gyp-bin/:"${PATH}"
    fi
```

Building node-rng

```
    $ cd node-rng
    $ node-gyp configure
    $ node-gyp build
```

Testing node-rng

- General usage:

```
$ node node-rng-example.js
```

- Statistical testing with dieharder: (note that this takes a long time)

```
$ sudo apt-get install dieharder
$ node node-rng-test.js | dieharder -a -g 200
```