

0.6.3 / 2021-05-23

- Fix HKSCS encoding to prefer Big5 codes if both Big5 and HKSCS codes are possible (#264)

0.6.2 / 2020-07-08

- Support Uint8Array-s decoding without conversion to Buffers, plus fix an edge case.

0.6.1 / 2020-06-28

- Support Uint8Array-s directly when decoding (#246, by @gyzerok)
- Unify package.json version ranges to be strictly semver-compatible (#241)
- Fix minor issue in UTF-32 decoder's endianness detection code.

0.6.0 / 2020-06-08

- Updated 'gb18030' encoding to :2005 edition (see <https://github.com/whatwg/encoding/issues/22>).
- Removed `iconv.extendNodeEncodings()` mechanism. It was deprecated 5 years ago and didn't work in recent Node versions.
- Reworked Streaming API behavior in browser environments to fix #204. Streaming API will be excluded by default in browser packs, saving ~100Kb bundle size, unless enabled explicitly using `iconv.enableStreamingAPI(require('stream'))`.
- Updates to development environment & tests:
 - Added `./test/webpack` private package to test complex new use cases that need custom environment. It's tested as a separate job in Travis CI.
 - Updated generation code for the new EUC-KR index file format from Encoding Standard.
 - Removed `Buffer()` constructor in tests (#197 by @gabrielschulhof).

0.5.2 / 2020-06-08

- Added `iconv.getEncoder()` and `iconv.getDecoder()` methods to type-script definitions (#229).
- Fixed semver version to 6.1.2 to support Node 8.x (by @tanandara).
- Capped iconv version to 2.x as 3.x has dropped support for older Node versions.
- Switched from Istanbul to c8 for code coverage.

0.5.1 / 2020-01-18

- Added cp720 encoding (#221, by @kr-deps)
- (minor) Changed Changelog.md formatting to use h2.

0.5.0 / 2019-06-26

- Added UTF-32 encoding, both little-endian and big-endian variants (UTF-32LE, UTF32-BE). If endianness is not provided for decoding, it's deduced automatically from the stream using a heuristic similar to what we use in UTF-16. (great work in #216 by @kshetline)
- Several minor updates to README (#217 by @oldj, plus some more)
- Added Node versions 10 and 12 to Travis test harness.

0.4.24 / 2018-08-22

- Added MIK encoding (#196, by @Ivan-Kalatchev)

0.4.23 / 2018-05-07

- Fix deprecation warning in Node v10 due to the last usage of `new Buffer` (#185, by @felixbuenemann)
- Switched from `NodeBuffer` to `Buffer` in typings (#155 by @felixfbecker, #186 by @larssn)

0.4.22 / 2018-05-05

- Use older semver style for dependencies to be compatible with Node version 0.10 (#182, by @dougwilson)
- Fix tests to accomodate fixes in Node v10 (#182, by @dougwilson)

0.4.21 / 2018-04-06

- Fix encoding canonicalization (#156)
- Fix the paths in the "browser" field in package.json (#174 by @LMLB)
- Removed "contributors" section in package.json - see Git history instead.

0.4.20 / 2018-04-06

- Updated `new Buffer()` usages with recommended replacements as it's being deprecated in Node v10 (#176, #178 by @ChALkeR)

0.4.19 / 2017-09-09

- Fixed iso8859-1 codec regression in handling untranslatable characters (#162, caused by #147)
- Re-generated windows1255 codec, because it was updated in iconv project
- Fixed grammar in error message when iconv-lite is loaded with encoding other than utf8

0.4.18 / 2017-06-13

- Fixed CESU-8 regression in Node v8.

0.4.17 / 2017-04-22

- Updated typescript definition file to support Angular 2 AoT mode (#153 by @larssn)

0.4.16 / 2017-04-22

- Added support for React Native (#150)
- Changed iso8859-1 encoding to use internal 'binary' encoding, as it's the same thing (#147 by @mscdex)
- Fixed typo in Readme (#138 by @jiangzhuo)
- Fixed build for Node v6.10+ by making correct version comparison
- Added a warning if iconv-lite is loaded not as utf-8 (see #142)

0.4.15 / 2016-11-21

- Fixed typescript type definition (#137)

0.4.14 / 2016-11-20

- Preparation for v1.0
- Added Node v6 and latest Node versions to Travis CI test rig
- Deprecated Node v0.8 support
- Typescript typings (@larssn)
- Fix encoding of Euro character in GB 18030 (inspired by @lygstate)
- Add ms prefix to dbcs windows encodings (@rokoroku)

0.4.13 / 2015-10-01

- Fix silly mistake in deprecation notice.

0.4.12 / 2015-09-26

- Node v4 support:
 - Added CESU-8 decoding (#106)
 - Added deprecation notice for `extendNodeEncodings`
 - Added Travis tests for Node v4 and io.js latest (#105 by @Mithgol)

0.4.11 / 2015-07-03

- Added CESU-8 encoding.

0.4.10 / 2015-05-26

- Changed UTF-16 endianness heuristic to take into account any ASCII chars, not just spaces. This should minimize the importance of “default” endianness.

0.4.9 / 2015-05-24

- Streamlined BOM handling: strip BOM by default, add BOM when encoding if addBOM: true. Added docs to Readme.
- UTF16 now uses UTF16-LE by default.
- Fixed minor issue with big5 encoding.
- Added io.js testing on Travis; updated node-iconv version to test against. Now we just skip testing SBCS encodings that node-iconv doesn’t support.
- (internal refactoring) Updated codec interface to use classes.
- Use strict mode in all files.

0.4.8 / 2015-04-14

- added alias UNICODE-1-1-UTF-7 for UTF-7 encoding (#94)

0.4.7 / 2015-02-05

- stop official support of Node.js v0.8. Should still work, but no guarantees. reason: Packages needed for testing are hard to get on Travis CI.
- work in environment where Object.prototype is monkey patched with enumerable props (#89).

0.4.6 / 2015-01-12

- fix rare aliases of single-byte encodings (thanks @mscdex)
- double the timeout for dbcs tests to make them less flaky on travis

0.4.5 / 2014-11-20

- fix windows-31j and x-sjis encoding support (@nleush)
- minor fix: undefined variable reference when internal error happens

0.4.4 / 2014-07-16

- added encodings UTF-7 (RFC2152) and UTF-7-IMAP (RFC3501 Section 5.1.3)
- fixed streaming base64 encoding

0.4.3 / 2014-06-14

- added encodings UTF-16BE and UTF-16 with BOM

0.4.2 / 2014-06-12

- don't throw exception if `extendNodeEncodings()` is called more than once

0.4.1 / 2014-06-11

- codepage 808 added

0.4.0 / 2014-06-10

- code is rewritten from scratch
- all widespread encodings are supported
- streaming interface added
- browserify compatibility added
- (optional) extend core primitive encodings to make usage even simpler
- moved from vows to mocha as the testing framework