Releases

Binary releases can be downloaded manually at: https://github.com/denoland/deno/releases

We also have one-line install commands at: https://github.com/denoland/deno_install

1.20.5 / 2022.04.07

- feat(lsp/unstable): add experimental testing API (#13798)
- feat(lsp/unstable): support tasks in the config file (#14139)
- feat(unstable): add ref/unref to Listener (#13961)
- fix(cli/install): preserve compat flag (#14223)
- fix(ext/crypto): check extractable in exportKey (#14222)

1.20.4 / 2022.03.31

- fix(compile): follow redirects when resolving (#14161)
- fix(ext/fetch): extend deprecated fetch() overload with `string | Request` (#14134)
- fix(lsp): watch .jsonc files (#14135)
- fix(runtime/ops/signal.rs): Add Solaris signals (#13931)
- fix(task): handle `PATHEXT` with trailing semi-colon (#14140)
- perf: micro-optimize core.encode (#14120)

1.20.3 / 2022.03.25

- fix(ext/fetch): deprecate URL as the first arg in types (#14113)
- fix(ext/ffi): enforce unstable check on ops (#14115)
- fix(runtime): do not modify user provided `cmd` array in `Deno.run` (#14109)

1.20.2 / 2022.03.24

- feat(lsp): support deno.enablePaths setting (#13978)
- fix(bench): require --unstable flag in JavaScript (#14091)
- fix(test): don't error on missing op details (#14074)
- fix(compat): Changes an instance of collect::<Vec<_>>().join("") to collect::() (#14082)
- fix(tests): do not use global env vars in install tests (#14078)
- fix(ext/fetch): Connect async error stack with user code (#13899)
- fix(unstable): upgrade deno_task_shell to 0.2 (#14073)
- fix: upgrade to swc_ecmascript 0.137.0 (#14067)
- fix(fetch): Fix uncaught rejection panic with `WebAssembly.instantiateStreaming` (#13925)
- fix(core): variadic opSync/opAsync (#14062)
- fix(runtime): actually don't inherit runtime permissions (#14024)
- fix(ext/console): fix error with a Proxy of a Map (#14032)
- fix(ops): throw TypeError on op return failure (#14033)
- fix(cli): improve `deno compile` error messages (#13944)
- fix(cli): add support for DENO_CERT in upgrade command (#13862)
- fix(config-file): fix config-file.v1.json schema to allow colons in the task name (#14013)
- perf(http): avoid Set.has() when closing connection resource (#14085)
- perf(http): avoid checking promise every request (#14079)
- perf(http): avoid per header alloc (#14051)

1.20.1 / 2022.03.16

- BREAKING: don't inherit permissions by default (#13668)

- feat(cli): support data url (#13667)
- feat(cli): update to TypeScript 4.6.2 (#13474)
- feat(compat): CJS/ESM interoperability (#13553)
- feat(core): Event loop middlewares for Extensions (#13816)
- feat(core): codegen ops (#13861)
- feat(ext/crypto): AES-GCM support for 128bit IVs (#13805)
- feat(ext/fetch): Allow Response status 101 (#13969)
- feat(ext/http): auto-compression of fixed response bodies (#13769)
- feat(ext/net): Use socket2 crate to create TcpListener (#13808)
- feat(ext/net): support cert, key options in listenTls (#13740)
- feat(ext/web): Add `AbortSignal.timeout()` (#13687)
- feat(net): add Deno.UnixConn interface (#13787)
- feat(ops): custom arity (#13949)
- feat(ops): optional OpState (#13954)
- feat(unstable): Add Deno.upgradeHttp API (#13618)
- feat: "deno bench" subcommand (#13713)
- feat: "deno task" subcommand (#13725)
- feat: Add Deno.TcpConn class, change return type from Deno.connect (#13714)
- feat: allow specification of import map in config file (#13739)
- feat: deno test --trace-ops (#13770)
- fix(compat): cjs/esm interop for dynamic imports (#13792)
- fix(core): Don't override structured clone error messages from V8 (#13942)
- fix(core): nuke Deno.core.ops pre-snapshot (#13970)
- fix(ext/crypto): handle JWK import with "use" (#13912)
- fix(ext/crypto): use EcKeyImportParams dictionary (#13894)
- fix(ext/http): drop content-length header on compression (#13866)
- fix(info): print deno info paths with unescaped backslashes on windows (#13847)
- fix(test): skip typechecking for blocks inside HTML comments (#13889)
- fix: shell completion hints (#13876)
- fix: upgrade reqwest to 0.11.10 (#13951)
- perf(web): Optimize `TextDecoder` by adding a new `U16String` type (#13923)
- perf(web): optimize Blob.text and Blob.arrayBuffer (#13981)
- perf(web): use DOMString for BlobParts (#13979)
- perf: opt-level-3 all of ext (#13940)

Note 1.20.0 was dead on arrival, see https://github.com/denoland/deno/pull/13993

1.19.3 / 2022.03.10

- fix(ci): restore compatibility with older glibc (#13846)
- fix(test): typecheck blocks annotated with long js/ts notations (#13785)
- perf(core): micro-optimize OpsTracker (#13868)
- perf(ext/web): optimize atob/btoa (#13841)
- perf(serde_v8): avoid SerializablePkg allocs (#13860)
- perf(serde_v8): optimize ByteString deserialization (#13853)

1.19.2 / 2022.03.03

- fix(cli): disable config discovery for remote script (#13745)
- fix(repl): fix null eval result (#13804)
- fix(runtime): disable console color for non tty stdout (#13782)
- fix(test): use --no-prompt by default (#13777)

1.19.1 / 2022.02.24

- feat(ext/ffi): Support read only global statics (#13662)
- fix(compile): Support import maps (#13756)
- fix(upgrade): move the file permission check to the beginning of the upgrade process (#13726)
- fix(vendor): do not add absolute specifiers to scopes (#13710)

1.19.0 / 2022.02.17

- feat: Add Deno.FsFile, deprecate Deno.File (#13660)
- feat: Add hint to permission prompt to display allow flag (#13695)
- feat: deno vendor (#13670)
- feat: never prompt for hrtime permission (#13696)
- feat: permission prompt by default (#13650)
- feat(compat): support --compat in web workers (#13629)
- feat(compile): Replace bundling with eszip in deno compile (#13563)
- feat(coverage): add "--output" flag (#13289)
- feat(ext/console): better circular information in object inspection (#13555)
- feat(ext/http): add support for unix domain sockets (#13628)
- feat(ext/net): Add Conn.setNoDelay and Conn.setKeepAlive (#13103)
- feat(ext/web): add CompressionStream API (#11728)
- feat(lsp): add redirect diagnostic and quick fix (#13580)
- feat(lsp): provide completions from import map if available (#13624)
- feat(lsp): support linking to symbols in JSDoc on hover (#13631)
- feat(runtime): stabilize addSignalListener API (#13438)
- feat(runtime): web streams in fs & net APIs (#13615)
- feat(test): better errors for resource sanitizer (#13296)
- feat(test): improved op sanitizer errors + traces (#13676)
- feat(watch): add "--no-clear-screen" flag (#13454)
- fix(compat): ESM resolver for package subpath (#13599)
- fix(ext/console): fix uncaught TypeError in css styling (#13567)
- fix(ext/console): print circular ref indicator in cyan (#13684)
- fix(ext/crypto): optional additionalData in encrypt/decrypt (#13669)
- fix(ext/crypto): support EC p256 private key material in exportKey (#13547)
- fix(lsp): do not panic getting root_uri to auto discover configuration file (#13603)
- fix(lsp): independent diagnostic publishing should include all diagnostic sources on each publish (#13483)
- fix(lsp): op_exists handles bad specifiers (#13612)

1.18.2 / 2022.02.03

- feat(unstable): add Deno.getUid (#13496)
- fix: don't crash when $HOME is a relative path (#13581)
- fix(cli): handle extensionless imports better (#13548)
- fix(cli): handle local files with query params on emit (#13568)
- fix(cli/dts/webgpu): make GPUBlendComponent properties optional (#13574)
- fix(ext/crypto): enforce 128bits tagLength for AES-GCM decryption (#13536)
- fix(ext/crypto): utf16 jwk encoding (#13535)
- fix(lsp): properly display x-deno-warning with redirects (#13554)
- fix(lsp): regression where certain diagnostics were showing for disabled files (#13530)
- fix(repl): tab completions (#13540)
- perf(lsp): cancellable TS diagnostics (#13565)

1.18.1 / 2022.01.27

- feat(unstable): add Deno.networkInterfaces (#13475)
- fix(ext/crypto): duplicate RsaHashedImportParams types (#13466)
- fix(lsp): respect DENO_CERT and other options related to TLS certs (#13467)
- perf(lsp): improve some tsc op hot paths (#13473)
- perf(lsp): independent diagnostic source publishes (#13427)

1.18.0 / 2022.01.20

- feat: auto-discover config file (#13313)
- feat: output `cause` on JS runtime errors (#13209)
- feat: stabilize test steps API (#13400)
- feat(cli, runtime): compress snapshots (#13320)
- feat(cli): add ignore directives to bundled code (#13309)
- feat(compat) preload Node.js built-in modules in global vars REPL (#13127)
- feat(ext/crypto): implement AES-GCM decryption (#13319)
- feat(ext/crypto): implement AES-GCM encryption (#13119)
- feat(ext/crypto): implement AES-KW for wrapKey/unwrapKey (#13286)
- feat(ext/crypto): implement pkcs8/JWK for P-384 curves (#13154)
- feat(ext/crypto): implement pkcs8/spki/jwk exportKey for ECDSA and ECDH (#13104)
- feat(ext/crypto): JWK support for unwrapKey/wrapKey (#13261)
- feat(ext/crypto): support AES-CTR encrypt/decrypt (#13177)
- feat(ext/crypto): support importing raw EC keys (#13079)
- feat(ext/ffi): infer symbol types (#13221)
- feat(ext/ffi): support alias names for symbol definitions (#13090)
- feat(ext/ffi): UnsafeFnPointer API (#13340)
- feat(ext/websocket): add header support to WebSocketStream (#11887)
- feat(ext/websocket): server automatically handle ping/pong for incoming WebSocket (#13172)
- feat(lsp): provide registry details on hover if present (#13294)
- feat(runtime): add op_network_interfaces (#12964)
- feat(serde_v8): deserialize ArrayBuffers (#13436)
- feat(streams): reject pending reads when releasing reader (#13375)
- feat(test): Add support for "deno test --compat" (#13235)
- fix(cli): Don't strip shebangs from modules (#13220)
- fix(cli): fix `deno install --prompt` (#13349)
- fix(cli/dts): add NotSupported error type (#13432)
- fix(ext/console): don't depend on globalThis present (#13387)
- fix(ext/crypto): validate maskGenAlgorithm asn1 in importKey (#13421)
- fix(ext/ffi): `pointer` type can accept `null` (#13335)
- fix(fmt): markdown formatting should not remove backslashed backslash at start of paragraph (#13429)
- fix(lsp): better handling of registry config errors (#13418)
- fix(runtime): don't crash when window is deleted (#13392)
- fix(streams): update TypeError message for pending reads when releasing reader (#13376)
- fix(tsc): Add typings for `Intl.ListFormat` (#13301)

1.17.3 / 2022.01.12

- fix: Get lib.deno_core.d.ts to parse correctly (#13238)
- fix: expose "Deno.memoryUsage()" in worker context (#13293)
- fix: install shim with `--allow-all` should not output each permission individually (#13325)
- fix(compile): fix output flag behaviour on compile command (#13299)
- fix(coverage): don't type check (#13324)
- fix(coverage): merge coverage ranges (#13334)

- fix(ext/web): handle no arguments in atob (#13341)
- fix(serde_v8): support #[serde(default)] (#13300)

1.17.2 / 2022.01.05

- fix(cli): include JSON modules in bundle (#13188)
- fix(core): inspector works if no "Runtime.runIfWaitingForDebugger" message is sent (#13191)
- fix(coverage): use only string byte indexes and 0-indexed line numbers (#13190)
- fix(doc): Make private types which show up in the rustdocs public (#13230)
- fix(ext/console): map basic css color keywords to ansi (#13175)
- fix(ext/crypto) - exportKey JWK for AES/HMAC must use base64url (#13264)
- fix(ext/crypto) include AES-CTR for deriveKey (#13174)
- fix(ext/crypto): use forgiving base64 encoding for JWK (#13240)
- fix(ext/ffi): throw errors instead of panic (#13283)
- fix(lsp): add code lens for tests just using named functions (#13218)
- fix(lsp): better handling of folders in registry completions (#13250)
- fix(lsp): handle repeating patterns in registry correctly (#13275)
- fix(lsp): properly generate data URLs for completion items (#13246)
- fix(signals): prevent panic when listening to forbidden signals (#13273)
- fix: support `mts`, `cjs` & `cts` files for `deno test` & `deno fmt` (#13274)
- fix: upgrade swc_ecmascript to 0.103 (#13284)

1.17.1 / 2021.12.22

- feat(lsp, unstable): add code lens for debugging tests (#13138)
- feat(lsp, unstable): supply accept header when fetching registry config (#13159)
- fix: inspector prompts (#13123)
- fix(coverage): Split sources by char index (#13114)
- fix(ext/ffi): use `c_char` instead of `i8` for reading strings (#13118)
- fix(ext/websocket): WebSocketStream don't error with "sending after closing" when closing (#13134)
- fix(repl): support assertions on import & export declarations (#13121)

1.17.0 / 2021.12.16

- feat: add `--no-check=remote` flag (#12766)
- feat: Add support for import assertions and JSON modules (#12866)
- feat: REPL import specifier auto-completions (#13078)
- feat: support abort reasons in Deno APIs and `WebSocketStream` (#13066)
- feat: support compat mode in REPL (#12882)
- feat(cli): update to TypeScript 4.5 (#12410)
- feat(core): Add ability to "ref" and "unref" pending ops (#12889)
- feat(core): intercept unhandled promise rejections (#12910)
- feat(ext/crypto): implement unwrapKey (#12539)
- feat(ext/crypto): support `importKey` in SPKI format (#12921)
- feat(ext/crypto): support exporting RSA JWKs (#13081)
- feat(ext/crypto): support importing ECSDA and ECDH (#13088)
- feat(ext/crypto): support importing exporting AES JWK keys (#12444)
- feat(ext/crypto): support importing RSA JWKs (#13071)
- feat(ext/fetch): Support `WebAssembly.instantiateStreaming` for file fetches (#12901)
- feat(ext/fetch): support abort reasons in fetch (#13106)
- feat(ext/ffi): implement UnsafePointer and UnsafePointerView (#12828)
- feat(ext/net): ALPN support in `Deno.connectTls()` (#12786)
- feat(ext/net): enable sending to broadcast address (#12860)

- feat(ext/timers): add refTimer, unrefTimer API (#12953)
- feat(ext/web): implement `AbortSignal.prototype.throwIfAborted()` (#13044)
- feat(lsp): add type definition provider (#12789)
- feat(lsp): add workspace symbol provider (#12787)
- feat(lsp): improve registry completion suggestions (#13023)
- feat(lsp): registry suggestion cache respects cache headers (#13010)
- feat(repl): add --unsafe-ignore-certificate-errors flag (#13045)
- feat(runtime): add op_set_exit_code (#12911)
- feat(streams): support abort reasons in streams (#12991)
- feat(test): Add more overloads for "Deno.test" (#12749)
- feat(watch): clear screen on each restart (#12613)
- feat(watch): support watching external files (#13087)
- fix: support "other" event type in FSWatcher (#12836)
- fix(cli): config file should resolve paths relative to the config file (#12867)
- fix(cli): don't add colors for non-tty outputs (#13031)
- fix(cli): don't cache .tsbuildinfo unless emitting (#12830)
- fix(cli): fix slow test, unbreak ci (#12897)
- fix(cli): skip bundling for pre-bundled code in "compile" (#12687)
- fix(ext/crypto): throw on key & op algo mismatch (#12838)
- fix(ext/crypto): various cleanup in JWK imports (#13092)
- fix(ext/net): make unix and tcp identical on close (#13075)
- fix(ext/timers): fix flakiness of `httpConnAutoCloseDelayedOnUpgrade` test (#13017)
- fix(ext/web): set location undefined when `--location` is not specified (#13046)
- fix(lsp): handle import specifier not having a trailing quote (#13074)
- fix(lsp): lsp should respect include/exclude files in format config (#12876)
- fix(lsp): normalize urls in did_change_watched_files (#12873)
- fix(lsp): provide diagnostics for import assertions (#13105)
- fix(workers): Make `worker.terminate()` not immediately kill the isolate (#12831)

1.16.4 / 2021.12.03

- fix(core): Wake up the runtime if there are ticks scheduled (#12933)
- fix(core): throw on invalid callConsole args (#12973) (#12974)
- fix(ext/crypto): throw on key & op algo mismatch (#12838)
- fix(test): Improve reliability of `deno test` 's op sanitizer with timers (#12934)
- fix(websocket): bad rid on WebSocketStream abort (#12913)
- fix(workers): Make `worker.terminate()` not immediately kill the isolate (#12831)

1.16.3 / 2021.11.24

- fix(cli): config file should resolve paths relative to the config file (#12867)
- fix(cli): don't cache .tsbuildinfo unless emitting (#12830)
- fix(cli/compile): skip bundling for pre-bundled code (#12687)
- fix(core): don't panic when evaluating module after termination (#12833)
- fix(core): keep event loop alive if there are ticks scheduled (#12814)
- fix(ext/crypto): don't panic on decryption failure (#12840)
- fix(ext/fetch): HTTP/1.x header case got discarded on the wire (#12837)
- fix(fmt): markdown formatting was incorrectly removing some non-breaking space html entities (#12818)
- fix(lsp): lsp should respect include/exclude files in format config (#12876)
- fix(lsp): normalize urls in did_change_watched_files (#12873)
- fix(lsp): tag deprecated diagnostics properly (#12801)
- fix(lsp): use lint exclude files list from the config file (#12825)

- fix(runtime): support "other" event type in FSWatcher (#12836)
- fix(runtime): support reading /proc using readFile (#12839)
- fix(test): do not throw on error.errors.map (#12810)

1.16.2 / 2021.11.17

- feat(unstable/test): include test step pass/fail/ignore counts in final report (#12432)
- fix(cli): short-circuit in prepare_module_load() (#12604)
- fix(lsp): retain module dependencies when parse is invalid (#12782)
- fix(test): support typechecking docs with CRLF line endings (#12748)
- fix(transpile): do not panic on `swc_ecma_utils::HANDLER` diagnostics (#12773)

1.16.1 / 2021.11.11

- feat(core): streams (#12596)
- fix(crypto): handling large key length in HKDF (#12692)
- fix: add typings for AbortSignal.reason (#12730)
- fix(http): non ascii bytes in response (#12728)
- fix: update unstable Deno props for signal API (#12723)

1.16.0 / 2021.11.09

- BREAKING(ext/web): remove `ReadableStream.getIterator` (#12652)
- feat(cli): support React 17 JSX transforms (#12631)
- feat(compat): add .code to dyn import error (#12633)
- feat(compat): integrate import map and classic resolutions in ESM resolution (#12549)
- feat(ext/console): Display error.cause in console (#12462)
- feat(ext/fetch): support fetching local files (#12545)
- feat(ext/net): add TlsConn.handshake() (#12467)
- feat(ext/web): BYOB support for ReadableStream (#12616)
- feat(ext/web): WritableStreamDefaultController.signal (#12654)
- feat(ext/web): add `AbortSignal.reason` (#12697)
- feat(ext/webstorage): use implied origin when --location not set (#12548)
- feat(runtime): add Deno.addSignalListener API (#12512)
- feat(runtime): give OS errors .code attributes (#12591)
- feat(test): better formatting for test elapsed time (#12610)
- feat(runtime): Stabilize Deno.TestDefinition.permissions (#12078)
- feat(runtime): stabilize Deno.startTls (#12581)
- feat(core): update to V8 9.7 (#12685)
- fix(cli): do not cache emit when diagnostics present (#12541)
- fix(cli): don't panic when mapping unknown errors (#12659)
- fix(cli): lint/format all discoverd files on each change (#12518)
- fix(cli): linter/formater watches current directory without args (#12550)
- fix(cli): no-check respects inlineSources compiler option (#12559)
- fix(cli/upgrade): nice error when unzip is missing (#12693)
- fix(encoding): support additional encoding labels (#12586)
- fix(ext/fetch): Replace redundant local variable with inline return statement (#12583)
- fix(ext/http): allow multiple values in upgrade header for websocket (#12551)
- fix(ext/net): expose all tls ops (#12699)
- fix(fetch): set content-length for empty POST/PUT (#12703)
- fix(fmt): reduce likelihood of deno fmt panic for file with multi-byte chars (#12623)
- fix(fmt/lint): strip unc paths on Windows when displaying file paths in lint and fmt (#12606)
- fix(lint): use recommended tag if there is no tags in config file or flags (#12644)

- fix(lint): use recommended tags when no tags specified in config, but includes or excludes are (#12700)
- fix(lsp): cache unsupported import completion origins (#12661)
- fix(lsp): display module types only dependencies on hover (#12683)
- fix(lsp): display signature docs as markdown (#12636)
- fix(runtime): require full read and write permissions to create symlinks (#12554)
- fix(tls): Make TLS clients support HTTP/2 (#12530)
- fix(webidl): Don't throw when converting a detached buffer source (#12585)
- fix(workers): Make `importScripts()` use the same HTTP client as `fetch` (#12540)
- fix: Deno.emit crashes with BorrowMutError (#12627)
- fix: support verbatim UNC prefixed paths on Windows (#12438)
- fix: typings for BYOB stream readers (#12651)
- perf(core): optimize waker capture in AsyncRefCell (#12332)
- perf(encoding): avoid copying the input data in `TextDecoder` (#12573)
- perf(http): encode string bodies in op-layer (#12451)
- perf: optimize some important crates more aggressively (#12332)

1.15.3 / 2021.10.25

- feat(serde_v8): StringOrBuffer (#12503)
- feat(serde_v8): allow all values to deserialize to unit type (#12504)
- fix(cli/dts): update std links for deprecations (#12496)
- fix(cli/tests): flaky Deno.watchFs() tests (#12485)
- fix(core): avoid op_state.borrow_mut() for OpsTracker (#12525)
- fix(core/bindings): use is_instance_of_error() instead of is_native_error() (#12479)
- fix(ext/net): fix TLS bugs and add 'op_tls_handshake' (#12501)
- fix(ext/websocket): prevent 'closed normally' panic (#12437)
- fix(lsp): formatting should error on certain additional swc diagnostics (#12491)
- fix: declare web types as global (#12497)

1.15.2 / 2021.10.18

- feat(unstable): Node CJS and ESM resolvers for compat mode (#12424)
- fix(cli): re-enable allowSyntheticDefaultImports for tsc (#12435)
- fix(cli/fmt_errors): don't panic on source line formatting errors (#12449)
- fix(cli/tests): move worker test assertions out of message handlers (#12439)
- fix(console): fix display of primitive wrapper objects (#12425)
- fix(core): avoid polling future after cancellation (#12385)
- fix(core): poll async ops eagerly (#12385)
- fix(fmt): keep parens for JS doc type assertions (#12475)
- fix(fmt): should not remove parens around sequence expressions (#12461)
- fix(runtime/ops/worker_host): move permission arg parsing to Rust (#12297)

1.15.1 / 2021.10.13

- fix: `--no-check` not properly handling code nested in TS expressions (#12416)
- fix: bundler panic when encountering export specifier with an alias (#12418)

1.15.0 / 2021.10.12

- feat: add --compat flag to provide built-in Node modules (#12293)
- feat: provide ops details for ops sanitizer failures (#12188)
- feat: Show the URL of streaming WASM modules in stack traces (#12268)
- feat: Stabilize Deno.kill and Deno.Process.kill (#12375)
- feat: stabilize Deno.resolveDns (#12368)

- feat: stabilize URLPattern API (#12256)
- feat: support serializing `WebAssembly.Module` objects (#12140)
- feat(cli/uninstall): add uninstall command (#12209)
- feat(ext/crypto): decode RSAES-OAEP-params with default values (#12292)
- feat(ext/crypto): export spki for RSA (#12114)
- feat(ext/crypto): implement AES-CBC encryption & decryption (#12123)
- feat(ext/crypto): implement deriveBits for ECDH (p256) (#11873)
- feat(ext/crypto): implement deriveKey (#12117)
- feat(ext/crypto): implement wrapKey (#12125)
- feat(ext/crypto): support importing raw ECDSA keys (#11871)
- feat(ext/crypto): support importing/exporting raw AES keys (#12392)
- feat(ext/ffi): add support for buffer arguments (#12335)
- feat(ext/ffi): Non-blocking FFI (#12274)
- feat(ext/net): relevant errors for resolveDns (#12370)
- feat(lint): add support for --watch flag (#11983)
- feat(runtime): allow passing extensions via Worker options (#12362)
- feat(runtime): improve error messages of runtime fs (#11984)
- feat(tls): custom in memory CA certificates (#12219)
- feat(unstable/test): imperative test steps API (#12190)
- feat(web): Implement `DOMException`'s `stack` property. (#12294)
- fix: Don't panic when a worker is closed in the reactions to a wasm operation. (#12270)
- fix: worker environment permissions should accept an array (#12250)
- fix(core/runtime): sync_ops_cache if nuked Deno ns (#12302)
- fix(ext/crypto): decode id-RSASSA-PSS with default params (#12147)
- fix(ext/crypto): key generation based on AES key length (#12146)
- fix(ext/crypto): missing Aes key typings (#12307)
- fix(ext/crypto): use NotSupportedError for importKey() (#12289)
- fix(ext/fetch): avoid panic when header is invalid (#12244)
- fix(ext/ffi): don't panic in dlopen (#12344)
- fix(ext/ffi): formatting dlopen errors on Windows (#12301)
- fix(ext/ffi): missing "buffer" type definitions (#12371)
- fix(ext/ffi): types for nonblocking FFI (#12345)
- fix(ext/http): merge identical if/else branches (#12269)
- fix(ext/net): should not panic when listening to unix abstract address (#12300)
- fix(ext/web): Format DOMException stack property (#12333)
- fix(http): don't expose body on GET/HEAD requests (#12260)
- fix(lsp): lint diagnostics respect config file (#12338)
- fix(repl): avoid panic when assigned to globalThis (#12273)
- fix(runtime): Declare `Window.self` and `DedicatedWorkerGlobalScope.name` with `util.writable()` (#12378)
- fix(runtime): don't equate SIGINT to SIGKILL on Windows (#12356)
- fix(runtime): Getting `navigator.hardwareConcurrency` on workers shouldn't throw (#12354)
- fix(runtime/js/workers): throw errors instead of using an op (#12249)
- fix(runtime/testing): format aggregate errors (#12183)
- perf(core): use opcall() directly (#12310)
- perf(fetch): fast path Uint8Array in extractBody() (#12351)
- perf(fetch): optimize fillHeaders() key iteration (#12287)
- perf(web): ~400x faster http header trimming (#12277)
- perf(web): optimize byteLowerCase() (#12282)
- perf(web/Event): move last class field to constructor (#12265)

- perf(webidl): fix typo from #12286 (#12336)
- perf(webidl): inline ResponseInit converter (#12285)
- perf(webidl): optimize createDictionaryConverter() (#12279)
- perf(webidl): optimize createRecordConverter() (#12286)
- perf(webidl/DOMString): don't wrap string primitives (#12266)

1.14.3 / 2021.10.04

- feat(core): implement Deno.core.isProxy() (#12288)
- fix(core/runtime): sync_ops_cache if nuked Deno ns (#12302)
- fix(ext/crypto): decode id-RSASSA-PSS with default params (#12147)
- fix(ext/crypto): missing Aes key typings (#12307)
- fix(ext/crypto): use NotSupportedError for importKey() (#12289)
- fix(ext/fetch): avoid panic when header is invalid (#12244)
- fix(ext/http): merge identical if/else branches (#12269)
- fix(ext/net): should not panic when listening to unix abstract address (#12300)
- fix(repl): avoid panic when assigned to globalThis (#12273)
- fix(runtime/js/workers): throw errors instead of using an op (#12249)
- fix(runtime/testing): format aggregate errors (#12183)
- fix: Don't panic when a worker is closed in the reactions to a wasm operation. (#12270)
- fix: worker environment permissions should accept an array (#12250)
- perf(core): use opcall() directly (#12310)
- perf(fetch): optimize fillHeaders() key iteration (#12287)
- perf(web): optimize byteLowerCase() (#12282)
- perf(web): ~400x faster http header trimming (#12277)
- perf(web/Event): move last class field to constructor (#12265)
- perf(webidl): optimize createDictionaryConverter() (#12279)
- perf(webidl): optimize createRecordConverter() (#12286)
- perf(webidl/DOMString): don't wrap string primitives (#12266)

1.14.2 / 2021.09.28

- feat(cli/fmt): support more markdown extensions (#12195)
- fix(cli/permissions): ensure revoked permissions are no longer granted (#12159)
- fix(ext/http): fortify "is websocket?" check (#12179)
- fix(ext/http): include port number in h2 urls (#12181)
- fix(ext/web): FileReader error messages (#12218)
- fix(ext/webidl): correctly apply [SymbolToStringTag] to interfaces (#11851)
- fix(http): panic when responding to a closed conn (#12216)
- fix(workers): Don't panic when a worker's parent thread stops running (#12156)
- fix: subprocess kill support on windows (#12134)
- perf(ext/fetch): Use the WebIDL conversion to DOMString rather than USVString for Response constructor (#12201)
- perf(ext/fetch): skip USVString webidl conv on string constructor (#12168)
- perf(fetch): optimize InnerBody constructor (#12232)
- perf(fetch): optimize newInnerRequest blob url check (#12245)
- perf(fetch/Response): avoid class fields (#12237)
- perf(fetch/headers): optimize appendHeader (#12234)
- perf(ops): optimize permission check (#11800)
- perf(web): optimize Event constructor (#12231)
- perf(webidl/ByteString): 3x faster ASCII check (#12230)
- quickfix(ci): only run "Build product size info" on main/tag (#12184)

- upgrade serde_v8 and rusty_v8 (#12175)

1.14.1 / 2021.09.21

- fix(cli): don't ignore diagnostics about for await (#12116)
- fix(cli): move Deno.flock and Deno.funlock to unstable types (#12138)
- fix(cli/fmt_errors): Abbreviate long data URLs in stack traces (#12127)
- fix(config-schema): correct default value of "lib" (#12145)
- fix(core): prevent multiple main module loading (#12128)
- fix(ext/crypto): don't use core.decode for encoding jwk keys (#12088)
- fix(ext/crypto): use DataError in importKey() (#12071)
- fix(lsp): align filter text to vscode logic (#12081)
- fix(runtime/ops/signal.rs): Add FreeBSD signal definitions (#12084)
- perf(ext/web): optimize EventTarget (#12166)
- perf(runtime/fs): optimize readFile by using a single large buffer (#12057)
- perf(web): optimize AbortController (#12165)

1.14.0 / 2021.09.14

- BREAKING(unstable): Fix casing in FfiPermissionDescriptor (#11659)
- BREAKING(unstable): Remove Deno.Signals enum, Deno.signals.* (#11909)
- feat(cli): Support Basic authentication in DENO_AUTH_TOKENS (#11910)
- feat(cli): Update to TypeScript 4.4 (#11678)
- feat(cli): add --ignore flag to test command (#11712)
- feat(cli): close test worker once all tests complete (#11727)
- feat(core): facilitate op-disabling middleware (#11858)
- feat(ext/crypto): AES key generation (#11869)
- feat(ext/crypto): export RSA keys as pkcs#8 (#11880)
- feat(ext/crypto): generate ECDH keys (#11870)
- feat(ext/crypto): implement HKDF operations (#11865)
- feat(ext/crypto): implement encrypt, decrypt & generateKey for RSA-OAEP (#11654)
- feat(ext/crypto): implement importKey and deriveBits for PBKDF2 (#11642)
- feat(ext/crypto): import RSA pkcs#8 keys (#11891)
- feat(ext/crypto): support JWK export for HMAC (#11864)
- feat(ext/crypto): support JWK import for HMAC (#11716)
- feat(ext/crypto): verify ECDSA signatures (#11739)
- feat(extensions/console): right align numeric columns in table (#11748)
- feat(fetch): mTLS client certificates for fetch() (#11721)
- feat(fmt): add basic JS doc formatting (#11902)
- feat(fmt): add support for configuration file (#11944)
- feat(lint): add support for config file and CLI flags for rules (#11776)
- feat(lsp): ignore specific lint for entire file (#12023)
- feat(unstable): Add file locking APIs (#11746)
- feat(unstable): Support file URLs in Deno.dlopen() (#11658)
- feat(unstable): allow specifing gid and uid for subprocess (#11586)
- feat(workers): Make the `Deno` namespace configurable and unfrozen (#11888)
- feat: ArrayBuffer in structured clone transfer (#11840)
- feat: add URLPattern API (#11941)
- feat: add option flags to 'deno fmt' (#12060)
- feat: stabilise Deno.upgradeWebSocket (#12024)
- fix(cli): better handling of source maps (#11954)
- fix(cli): dispatch unload event on watch drop (#11696)

- fix(cli): retain path based test mode inference (#11878)
- fix(cli): use updated names in deno info help text (#11989)
- fix(doc): fix rustdoc bare_urls warning (#11921)
- fix(ext/crypto): KeyAlgorithm typings for supported algorithms (#11738)
- fix(ext/crypto): add HkdfParams and Pkdf2Params types (#11991)
- fix(ext/fetch): Properly cancel upload stream when aborting (#11966)
- fix(ext/http): resource leak if request body is not consumed (#11955)
- fix(ext/http): websocket upgrade header check (#11830)
- fix(ext/web): Format terminal DOMExceptions properly (#11834)
- fix(ext/web): Preserve stack traces for DOMExceptions (#11959)
- fix(lsp): correctly parse registry patterns (#12063)
- fix(lsp): support data urls in `deno.importMap` option (#11397)
- fix(runtime): return error instead of panicking for windows signals (#11940)
- fix(test): propagate join errors in deno test (#11953)
- fix(typings): fix property name in DiagnosticMessageChain interface (#11821)
- fix(workers): don't drop messages from workers that have already been closed (#11913)
- fix: FileReader onevent attributes don't conform to spec (#11908)
- fix: FileReader.readAsText compat (#11814)
- fix: Query string percent-encoded in import map (#11976)
- fix: a `Request` whose URL is a revoked blob URL should still fetch (#11947)
- fix: bring back Deno.Signal to unstable props (#11945)
- fix: change assertion in httpServerIncompleteMessage test (#12052)
- fix: exit process on panic in a tokio task (#11942)
- fix: move unstable declarations to deno.unstable (#11876)
- fix: permission prompt stuffing (#11931)
- fix: permission prompt stuffing on Windows (#11969)
- fix: remove windows-only panic when calling `Deno.kill` (#11948)
- fix: worker_message_before_close was flaky (#12019)
- perf(ext/http): optimize auto cleanup of request resource (#11978)

Release notes for std version 0.107.0: https://github.com/denoland/deno_std/releases/tag/0.107.0

1.13.2 / 2021.08.23

- fix(cli/flags): require a non zero usize for concurrent jobs (#11802)
- fix(ext/crypto): exportKey() for HMAC (#11737)
- fix(ext/crypto): remove duplicate Algorithm interface definition (#11807)
- fix(ext/ffi): don't panic on invalid enum values (#11815)
- fix(ext/http): resource leak on HttpConn.close() (#11805)
- fix(lsp): better handling of languageId (#11755)
- fix(runtime): event loop panics in classic workers (#11756)
- fix(ext/fetch): Headers constructor error message (#11778)
- perf(ext/url): cleanup and optimize url parsing op args (#11763)
- perf(ext/url): optimize UrlParts op serialization (#11765)
- perf(ext/url): use DOMString instead of USVString as webidl converter for URL parsing (#11775)
- perf(url): build with opt-level 3 (#11779)

Release notes for std version 0.106.0: https://github.com/denoland/deno_std/releases/tag/0.106.0

1.13.1 / 2021.08.16

- fix: Blob#slice arguments should be optional (#11665)

- fix: correct spelling of certificate in `--unsafely-ignore-certificate-errors` warning message (#11634)
- fix: don't statically type name on Deno.errors (#11715)
- fix: parse error when transpiling code with BOM (#11688)
- fix(cli): allow specifiers of unknown media types with test command (#11652)
- fix(cli): explicitly scan for ignore attribute in inline tests (#11647)
- fix(cli): retain input order of remote specifiers (#11700)
- fix(cli/lint): don't use gray in diagnostics output for visibility (#11702)
- fix(cli/tools/repl): don't highlight candidate when completion is list (#11697)
- fix(ext/crypto): enable non-extractable keys (#11705)
- fix(ext/crypto): fix copying buffersource (#11714)
- fix(ext/crypto): handle idlValue not being present (#11685)
- fix(ext/crypto): importKey() SecurityError on non-extractable keys (#11662)
- fix(ext/crypto): take a copy of keyData bytes (#11666)
- fix(ext/fetch): better error if no content-type
- fix(ext/fetch): don't use global Deno object
- fix(ext/http): remove unwrap() when HTTP conn errors (#11674)
- fix(ext/web): use Array primordials in MessagePort (#11680)
- fix(http/ws): support multiple options in connection header (#11675)
- fix(lint): add links to help at lint.deno.land (#11667)
- fix(test): dispatch load event before tests are run (#11708)
- fix(test): sort file module specifiers (#11656)
- perf: improve localStorage throughput (#11709)
- perf(ext/http): faster req_url string assembly (#11711)
- perf(wpt/crypto): optimize num-bigint-dig for debug builds (#11681)

Release notes for std version 0.105.0: https://github.com/denoland/deno_std/releases/tag/0.105.0

1.13.0 / 2021.08.10

- BREAKING(unstable): Rename Deno.WebSocketUpgrade::websocket to socket (#11542)
- feat: Add --unsafely-treat-insecure-origin-as-secure flag to disable SSL verification (#11324)
- feat: add experimental WebSocketStream API (#10365)
- feat: FFI API replacing native plugins (#11152)
- feat: stabilize Deno.serveHttp() (#11544)
- feat: support AbortSignal in writeFile (#11568)
- feat: support client certificates for connectTls (#11598)
- feat: type check codeblocks in Markdown file with "deno test --doc" (#11421)
- feat(extensions/crypto): implement importKey and exportKey for raw HMAC keys (#11367)
- feat(extensions/crypto): implement verify() for HMAC (#11387)
- feat(extensions/tls): Optionally support loading native certs (#11491)
- feat(extensions/web): add structuredClone function (#11572)
- feat(fmt): format top-level JSX elements/fragments with parens when multi-line (#11582)
- feat(lsp): ability to set DENO_DIR via settings (#11527)
- feat(lsp): implement refactoring code actions (#11555)
- feat(lsp): support clients which do not support disabled code actions (#11612)
- feat(repl): add --eval flag for evaluating code when the repl starts (#11590)
- feat(repl): support exports in the REPL (#11592)
- feat(runtime): allow URL for permissions (#11578)
- feat(runtime): implement navigator.hardwareConcurrency (#11448)
- feat(unstable): clean environmental variables for subprocess (#11571)
- fix: support windows file specifiers with import maps (#11551)

- fix: Type `Deno.errors.*` as subclasses of `Error` (#10702)
- fix(doc): panic on invalid url (#11536)
- fix(extensions/fetch): Add Origin header to outgoing requests for fetch (#11557)
- fix(extensions/websocket): allow any close code for server (#11614)
- fix(lsp): do not output to stderr before exiting the process (#11562)

Release notes for std version 0.104.0: https://github.com/denoland/deno_std/releases/tag/0.104.0

1.12.2 / 2021.07.26

- feat(lsp, unstable): add workspace config to status page (#11459)
- fix: panic for non-WS connections to inspector (#11466)
- fix: support --cert flag for TLS connect APIs (#11484)
- fix(cli): info now displays type reference deps (#11478)
- fix(cli): normalize test command errors (#11375)
- fix(cli): rebuild when environment variables change (#11471)
- fix(cli): side-load test modules (#11515)
- fix(extensions/fetch): close fetch response body on GC (#11467)
- fix(extensions/http): support multiple options in connection header for websocket (#11505)
- fix(extensions/websocket): case insensitive connection header (#11489)
- fix(lsp): do not populate maybe_type slot with import type dep (#11477)
- fix(lsp): handle importmaps properly (#11496)

Release notes for std version 0.103.0: https://github.com/denoland/deno_std/releases/tag/0.103.0

1.12.1 / 2021.07.19

- fix: Big{U|}Int64Array in crypto.getRandomValues (#11447)
- fix(extensions/http): correctly concat cookie headers (#11422)
- fix(extensions/web): aborting a FileReader should not affect later reads (#11381)
- fix(repl): output error without hanging when input is invalid (#11426)
- fix(tsc): add .at() types manually to tsc (#11443)
- fix(workers): silently ignore non-existent worker IDs (#11417)

Release notes for std version 0.102.0: https://github.com/denoland/deno_std/releases/tag/0.102.0

1.12.0 / 2021.07.13

- feat: Add `MessageChannel` and `MessagePort` APIs (#11051)
- feat: Deno namespace configurable and unfrozen (#11062)
- feat: Enable WebAssembly.instantiateStreaming and WebAssembly.compileStreaming (#11200)
- feat: Support "types" option when type checking (#10999)
- feat: Support SharedArrayBuffer sharing between workers (#11040)
- feat: Transfer MessagePort between workers (#11076)
- feat(extensions/crypto): Implement generateKey() and sign() (#9614)
- feat(extensions/crypto): Implement verify() for RSA (#11312)
- feat(extensions/fetch): Add programmatic proxy (#10907)
- feat(extensions/http): Server side websocket support (#10359)
- feat(inspector): Improve inspector prompt in Chrome Devtools (#11187)
- feat(inspector): Pipe console messages between terminal and inspector (#11134)
- feat(lsp): Dependency hover information (#11090)
- feat(repl): Show list completion (#11001)
- feat(repl): Support autocomplete on declarations containing a primitive (#11325)
- feat(repl): Support import declarations in the REPL (#11086)

- feat(repl): Type stripping in the REPL (#10934)
- feat(test): Add "--shuffle" flag to randomize test ordering (#11163)
- feat(test): Add support for "--fail-fast=N" (#11316)
- fix: Align DedicatedWorkerGlobalScope event handlers to spec (#11353)
- fix: Move stable/unstable types/APIs to their correct places (#10880)
- fix(core): Fix concurrent loading of dynamic imports (#11089)
- fix(extensions/console): Eliminate panic inspecting event classes (#10979)
- fix(extensions/console): Inspecting prototypes of built-ins with custom inspect implementations should not throw (#11308)
- fix(extensions/console): Left align table entries (#11295)
- fix(extensions/crypto): Hash input for RSASSA-PKCS1-v1_5 before signing (#11314)
- fix(extensions/fetch): Consumed body with a non-stream source should result in a disturbed stream (#11217)
- fix(extensions/fetch): Encode and decode headers as byte strings (#11070)
- fix(extensions/fetch): Filter out custom HOST headers (#11020)
- fix(extensions/fetch): OPTIONS should be allowed a non-null body (#11242)
- fix(extensions/fetch): Proxy body for requests created from other requests (#11093)
- fix(extensions/http): Encode and decode headers as byte strings in the HTTP server (#11144)
- fix(extensions/http): Panic in request body streaming (#11191)
- fix(extensions/http): Specify AbortSignal for native http requests (#11126)
- fix(extensions/timers): Spec conformance for performance API (#10887)
- fix(extensions/url): Use USVStrings in URLSearchParams constructor (#11101)
- fix(extensions/web): AddEventListenerOptions.signal shouldn't be nullable (#11348)
- fix(extensions/webgpu): Align error scopes to spec (#9797)
- fix(lsp): Handle invalid config setting better (#11104)
- fix(lsp): Reload import registries should not error when the module registries directory does not exist (#11123)
- fix(repl): Panic when Deno.inspect throws (#11292)
- fix(runtime): Fix signal promise API (#11069)
- fix(runtime): Ignored tests should not cause permission changes (#11278)

Release notes for std version 0.101.0: https://github.com/denoland/deno_std/releases/tag/0.101.0

1.11.3 / 2021.06.29

- fix(#10761): graph errors reported as diagnostics for `Deno.emit()` (#10767)
- fix(core): don't panic on stdout/stderr write failures in Deno.core.print (#11039)
- fix(core): top-level-await is now always enabled (#11082)
- fix(extensions/fetch): Filter out custom HOST headers (#11020)
- fix(fetch): proxy body for requests created from other requests (#11093)
- fix(http): remove unwrap() in HTTP bindings (#11130)
- fix(inspect): eliminate panic inspecting event classes (#10979)
- fix(lsp): reload import registries should not error when the module registries directory does not exist (#11123)
- fix(runtime): fix signal promise API (#11069)
- fix(runtime/signal): use op_async_unref for op_signal_poll (#11097)
- fix(url): use USVStrings in URLSearchParams constructor (#11101)
- fix(webstorage): increase localStorage limit to 10MB (#11081)
- fix: make readonly `Event` properties readonly (#11106)
- fix: specify AbortSignal for native http requests (#11126)
- chore: upgrade crates (#11007)

- chore: use lsp to get parent process id (#11083)

Release notes for std version 0.100.0: https://github.com/denoland/deno_std/releases/tag/0.100.0

1.11.2 / 2021.06.21

- feat(unstable, lsp): quick fix actions to ignore lint errors (#10627)
- fix: add support for module es2020 to Deno.emit (#11065)
- fix: align Console to spec (#10983)
- fix: align URL / URLSearchParams to spec (#11005)
- fix: align Websocket to spec (#11010)
- fix: closing / aborting WritableStream is racy (#10982)
- fix: fetch with method HEAD should not have body (#11003)
- fix: Worker accepts specifier as URL (#11038)
- fix(lsp): do not rename in strings and comments (#11041)

1.11.1 / 2021.06.15

- feat(unstable): add additional logging information in LSP (#10890)
- fix: Deno.inspect should inspect the object the proxy represents rather than the target of the proxy (#10977)
- fix: early binding to dispatchEvent in workers (#10904)
- fix: hang in Deno.serveHttp() (#10923)
- fix: improve worker types (#10965)
- fix: make WHATWG streams more compliant (#10967, #10970)
- fix: poll connection after writing response chunk in Deno.serveHttp() (#10961)
- fix: set minimum timeout to be 4 milliseconds (#10972)
- fix(repl): Complete declarations (#10963)
- fix(repl): Fix `undefined` result colour in cmd (#10964)

Release notes for std version 0.99.0: https://github.com/denoland/deno_std/releases/tag/0.99.0

1.11.0 / 2021.06.08

- feat: Add FsWatcher interface (#10798)
- feat: Add origin data dir to deno info (#10589)
- feat: Initialize runtime_compiler ops in `deno compile` (#10052)
- feat: Make 'deno lint' stable (#10851)
- feat: Support data uri dynamic imports in `deno compile` (#9936)
- feat: upgrade to TypeScript 4.3 (#9960)
- feat(extensions): add BroadcastChannel
- feat(extensions/crypto): implement randomUUID (#10848)
- feat(extensions/crypto): implement subtle.digest (#10796)
- feat(extensions/fetch): implement abort (#10863)
- feat(extensions/web): Implement TextDecoderStream and TextEncoderStream (#10842)
- feat(lsp): add test code lens (#10874)
- feat(lsp): registry auto discovery (#10813)
- fix: change Crypto to interface (#10853)
- fix: Support the stream option to TextDecoder#decode (#10805)
- fix(extensions/fetch): implement newline normalization and escapes in the multipart/form-data serializer (#10832)
- fix(runtime/http): Hang in `Deno.serveHttp` (#10836)
- fix(streams): expose ReadableByteStreamController & TransformStreamDefaultController (#10855)

Release notes for std version 0.98.0: https://github.com/denoland/deno_std/releases/tag/0.98.0

1.10.3 / 2021.05.31

- feat(lsp): diagnostics for deno types and triple-slash refs (#10699)
- feat(lsp): provide X-Deno-Warning as a diagnostic (#10680)
- feat(lsp): show hints from `deno_lint` in addition to messages (#10739)
- feat(lsp): support formatting json and markdown files (#10180)
- fix(cli): always allow documentation modules to be checked (#10581)
- fix(cli): canonicalize coverage dir (#10364)
- fix(cli): don't statically error on dynamic unmapped bare specifiers (#10618)
- fix(cli): empty tsconfig.json file does not cause error (#10734)
- fix(cli): support source maps with Deno.emit() and bundle (#10510)
- fix(cli/dts): fix missing error class (NotSupported) in types (#10713)
- fix(cli/install): support `file:` scheme URLs (#10562)
- fix(cli/test): don't use reserved symbol `:` in specifier (#10751)
- fix(cli/test): ensure coverage dir exists (#10717)
- fix(cli/upgrade): modify download size paddings (#10639)
- fix(runtime/http): expose nextRequest() errors in respondWith() (#10384)
- fix(runtime/http): fix empty blob response (#10689)
- fix(serde_v8): remove intentional deserialization error on non-utf8 strings (#10156)
- fix(ext/fetch): fix error message of Request constructor (#10772)
- fix(ext/fetch): make prototype properties writable (#10769)
- fix(ext/fetch): remove unimplemented Request attributes (#10784)
- fix(ext/file): update File constructor following the spec (#10760)
- fix(ext/webstorage): use opstate for sqlite connection (#10692)
- fix(lsp): deps diagnostics include data property (#10696)
- fix(lsp): ignore type definition not found diagnostic (#10610)
- fix(lsp): local module import added by code action now includes the file extension (#10778)
- fix(lsp): make failed to load config error descriptive (#10685)
- fix(lsp): memoize script versions per tsc request (#10601)
- fix(lsp): re-enable the per resource configuration without a deadlock (#10625)

1.10.2 / 2021.05.17

- fix: static import permissions in dynamic imports
- fix(lsp): remove duplicate cwd in config path (#10620)
- fix(cli): ignore x-typescript-types header when media type is not js/jsx (#10574)
- chore: upgrade Tokio to 1.6.0 (#10637)

Release notes for std version 0.97.0: https://github.com/denoland/deno_std/releases/tag/0.97.0

1.10.1 / 2021.05.11

- fix(#10603): Disable lsp workspaces, resolve deadlock bug

1.10.0 / 2021.05.11

- feat: "deno test" prompts number of tests and origin (#10428)
- feat: "Worker.postMessage()" uses structured clone algorithm (#9323)
- feat: add "deno test --doc" (#10521)
- feat: add "deno test --jobs" (#9815)
- feat: add "deno test --watch" (#9160)
- feat: add test permissions to Deno.test (#10188)
- feat: add WebStorage API (#7819)
- feat: align plugin api with "deno_core::Extension" (#10427)

- feat: support deno-fmt-ignore-file for markdown formatting (#10191)
- feat(core): enable WASM shared memory (#10116)
- feat(core): introduce Extension (#9800)
- feat(lsp): add internal debugging logging (#10438)
- feat(lsp): support workspace folders configuration (#10488)
- fix: invalid types for asynchronous and synchronous `File#truncate` (#10353)
- fix: rename Deno.emit() bundle options to "module" and "classic" (#10332)
- fix: sleepSync doesn't return a Promise (#10358)
- fix: TextEncoder#encodeInto spec compliance (#10129)
- fix: typings for `Deno.os.arch` (#10541)
- fix(extensions/fetch): infinite loop on fill headers (#10406)
- fix(extensions/fetch): Prevent throwing when inspecting a request (#10335)
- fix(installer): allow remote import maps (#10499)
- fix(lsp): remove code_action/diagnostics deadlock (#10555)
- fix(tls): flush send buffer in the background after closing TLS stream (#10146)
- fix(tls): throw meaningful error when hostname is invalid (#10387)

Release notes for std version 0.96.0: https://github.com/denoland/deno_std/releases/tag/0.96.0

1.9.2 / 2021.04.23

- fix: parse websocket messages correctly (#10318)
- fix: standalone bin corruption on M1 (#10311)
- fix: don't gray-out internal error frames (#10293)
- fix(op_crates/fetch): Response inspect regression (#10295)
- fix(runtime): do not panic on not found cwd (#10238)
- fix(op_crates/webgpu): move non-null op buffer arg check when needed (#10319)
- fix(lsp): document symbol performance mark (#10264)

Release notes for std version 0.95.0: https://github.com/denoland/deno_std/releases/tag/0.95.0

1.9.1 / 2021.04.20

- feat(lsp, unstable): Implement textDocument/documentSymbol (#9981)
- feat(lsp, unstable): implement textDocument/prepareCallHierarchy (#10061)
- feat(lsp, unstable): Implement textDocument/semanticTokens/full (#10233)
- feat(lsp, unstable): improve diagnostic status page (#10253)
- fix: revert changes to Deno.Conn type (#10255)
- fix(lsp): handle x-typescript-types header on type only imports properly (#10261)
- fix(lsp): remove documents when closed (#10254)
- fix(runtime): correct URL in Request (#10256)
- fix(runtime): handle race condition in postMessage where worker has terminated (#10239)
- fix(runtime): hang during HTTP server response (#10197)
- fix(runtime): include HTTP ops in WebWorker scope (#10207)

Release notes for std version 0.94.0: https://github.com/denoland/deno_std/releases/tag/0.94.0

1.9.0 / 2021.04.13

- feat: blob URL support (#10045)
- feat: blob URL support in fetch (#10120)
- feat: data URL support in fetch (#10054)
- feat: native HTTP bindings (#9935)
- feat: raise file descriptor limit on startup (#10162)

- feat: set useDefineForClassFields to true (#10119)
- feat: stabilize Deno.ftruncate and Deno.ftruncateSync (#10126)
- feat: stricter typings for Listener & Conn (#10012)
- feat(lsp): add import completions (#9821)
- feat(lsp): add registry import auto-complete (#9934)
- feat(lsp): implement textDocument/foldingRange (#9900)
- feat(lsp): implement textDocument/selectionRange (#9845)
- feat(permissions): allow env permission to take values (#9825)
- feat(permissions): allow run permission to take values (#9833)
- feat(runtime): add stat and statSync methods to Deno.File (#10107)
- feat(runtime): add truncate and truncateSync methods to Deno.File (#10130)
- feat(runtime): stabilize Deno.fstat and Deno.fstatSync (#10108)
- feat(runtime/permissions): prompt fallback (#9376)
- feat(unstable): Add Deno.memoryUsage() (#9986)
- feat(unstable): ALPN config in listenTls (#10065)
- fix: include deno.crypto in "deno types" (#9863)
- fix: Properly await already evaluating dynamic imports (#9984)
- fix(lsp): don't error on tsc debug failures for code actions (#10047)
- fix(lsp): ensure insert_text is passed back on completions (#9951)
- fix(lsp): folding range adjustment panic (#10030)
- fix(lsp): normalize windows file URLs properly (#10034)
- fix(lsp): properly handle encoding URLs from lsp client (#10033)
- fix(op_crates/console): console.table value misalignment with varying keys (#10127)
- fix(permissions): don't panic when no input is given (#9894)
- fix(runtime/js/timers): Use (0, eval) instead of eval() (#10103)
- fix(runtime/readFile*): close resources on error during read (#10059)
- fix(websocket): ignore resource close error (#9755)

Release notes for std version 0.93.0: https://github.com/denoland/deno_std/releases/tag/0.93.0

1.8.3 / 2021.04.02

- feat(lsp): add import completions (#9821)
- feat(lsp): implement textDocument/selectionRange (#9845)
- fix(websocket): ignore resource close error (#9755)
- fix(lsp): ensure insert_text is passed back on completions (#9951)
- fix(web): add AbortController.abort() (#9907)
- fix(crypto): include deno.crypto in `deno types` (#9863)
- fix(cli): re-add dom.asynciterable lib (#9888)

Release notes for std version 0.92.0: https://github.com/denoland/deno_std/releases/tag/0.92.0

1.8.2 / 2021.03.21

- fix: fallback to default UA and CA data for Deno.createHttpClient() (#9830)
- fix: getBindGroupLayout always illegal invocation (#9684)
- fix(cli/bundle): display anyhow error chain (#9822)
- fix(core): don't panic on invalid arguments for Deno.core.print (#9834)
- fix(doc): update example for sub processes (#9798)
- fix(fmt): Correctly format hard breaks in markdown (#9742)
- fix(lsp): allow on disk files to change (#9746)
- fix(lsp): diagnostics use own thread and debounce (#9572)
- fix(op_crates/webgpu): create instance only when required (#9771)

- fix(runtime): do not require deno namespace in workers for crypto (#9784)
- refactor: enforce type ResourceId across codebase (#9837, #9832)
- refactor: Clean up permission handling (#9367)
- refactor: Move bin ops to deno_core and unify logic with json ops (#9457)
- refactor: Move Console to op_crates/console (#9770)
- refactor: Split web op crate (#9635)
- refactor: Simplify icu data alignment (#9766)
- refactor: Update minimal ops & rename to buffer ops (#9719)
- refactor: Use serde ops more (#9817, #9828)
- refactor(lsp): refactor completions and add tests (#9789)
- refactor(lsp): slightly reorganize diagnostics debounce logic (#9796)
- upgrade: rusty_v8 0.21.0 (#9725)
- upgrade: tokio 1.4.0 (#9842)

Release notes for std version 0.91.0: https://github.com/denoland/deno_std/releases/tag/0.91.0

1.8.1 / 2021.03.09

- fix(cli/ast): Pass importsNotUsedAsValues to swc (#9714)
- fix(cli/compile): Do not append .exe depending on target (#9668)
- fix(cli/coverage): Ensure single line functions don't yield false positives (#9717)
- fix(core): Shared queue assertion failure in case of js error (#9721)
- fix(runtime): Add navigator interface objects (#9685)
- fix(runtime/web_worker): Don't block self.onmessage with TLA (#9619)
- fix(webgpu): Add Uint32Array type for code in ShaderModuleDescriptor (#9730)
- fix(webgpu): Add webidl records and simple unions (#9698)

Release notes for std version 0.90.0: https://github.com/denoland/deno_std/releases/tag/0.90.0

1.8.0 / 2021.03.02

https://deno.land/posts/v1.8

- feat: Align import map to spec and stabilize (#9616, #9526)
- feat: Deno.emit supports bundling as IIFE (#9291)
- feat: Use top user frame for error source lines (#9604)
- feat: WebGPU API (#7977)
- feat: add "deno coverage" subcommand (#8664)
- feat: add --ext flag to deno eval (#9295)
- feat: add exit sanitizer to Deno.test (#9529)
- feat: add json(c) support to deno fmt (#9292)
- feat: add structured cloning to Deno.core (#9458)
- feat: per op metrics (unstable) (#9240)
- feat: represent type dependencies in info (#9630)
- feat: stabilize Deno.permissions (#9573)
- feat: stabilize Deno.link and Deno.linkSync (#9417)
- feat: stabilize Deno.symlink and Deno.symlinkSync (#9226)
- feat: support auth tokens for accessing private modules (#9508)
- feat: support loading import map from URL (#9519)
- feat: use type definitions "deno doc" if available (#8459)
- fix(core): Add stacks for dynamic import resolution errors (#9562)
- fix(core): Fix dynamic imports for already rejected modules (#9559)
- fix(lsp): improve exception handling on tsc snapshots (#9628)

- fix(repl): filter out symbol candidates (#9555)
- fix(runtime): do not panic on irregular dir entries (#9579)
- fix(runtime/testing): false positive for timers when an error is thrown (#9553)
- fix(websocket): default to close code 1005 (#9339)
- fix: lint and fmt error if no target files are found (#9527)
- fix: panic caused by Deno.env.set("", "") (#9583)
- fix: typo in coverage exit_unstable (#9626)
- upgrade: TypeScript 4.2 (#9341)
- upgrade: rusty_v8 (V8 9.0.257.3) (#9605)

Release notes for std version 0.89.0: https://github.com/denoland/deno_std/releases/tag/0.89.0

1.7.5 / 2021.02.19

- fix: align btoa to spec (#9053)
- fix: Don't use file names from source maps (#9462)
- fix: Make dynamic import async errors catchable (#9505)
- fix: webidl utils and align `Event` to spec (#9470)
- fix(lsp): document spans use original range (#9525)
- fix(lsp): handle cached type dependencies properly (#9500)
- fix(lsp): handle data URLs properly (#9522)

Release notes for std version 0.88.0: https://github.com/denoland/deno_std/releases/tag/0.88.0

1.7.4 / 2021.02.13

- feat(unstable, lsp): add deno cache code actions (#9471)
- feat(unstable, lsp): add implementations code lens (#9441)
- fix(cli): check for inline source maps before external ones (#9394)
- fix(cli): fix WebSocket close (#8776)
- fix(cli): import maps handles data URLs (#9437)
- fix(console): log function object properties / do not log non-enumerable props by default (#9363)
- fix(lsp): handle code lenses for non-documents (#9454)
- fix(lsp): handle type deps properly (#9436)
- fix(lsp): prepare diagnostics when the config changes (#9438)
- fix(lsp): properly handle static assets (#9476)
- fix(lsp): support codeAction/resolve (#9405)
- fix(op_crates): Don't use `Deno.inspect` in op crates (#9332)
- fix(runtime/tls): handle invalid host for connectTls/startTls (#9453)
- upgrade: rusty_v8 0.17.0, v8 9.0.123 (#9413)
- upgrade: deno_doc, deno_lint, dprint, swc_ecmascript, swc_bundler (#9474)

Release notes for std version 0.87.0: https://github.com/denoland/deno_std/releases/tag/0.87.0

v1.7.3 was released but quickly removed due to bug #9484.

1.7.2 / 2021.02.05

- feat(lsp, unstable): add references code lens (#9316)
- feat(lsp, unstable): add TS quick fix code actions (#9396)
- fix: improve http client builder error message (#9380)
- fix(cli): fix handling of non-normalized specifier (#9357)
- fix(cli/coverage): display mapped instrumentation line counts (#9310)
- fix(cli/lsp): fix using jsx/tsx when not emitting via tsc (#9407)

- fix(repl): prevent symbol completion panic (#9400)
- refactor: rewrite Blob implementation (#9309)
- refactor: rewrite File implementation (#9334)

Release notes for std version 0.86.0: https://github.com/denoland/deno_std/releases/tag/0.86.0

1.7.1 / 2021.01.29

- feat(lsp, unstable): add performance measurements (#9209)
- fix(cli): IO resource types, fix concurrent read/write and graceful close (#9118)
- fix(cli): Move WorkerOptions::deno types to unstable (#9163)
- fix(cli): add lib dom.asynciterable (#9288)
- fix(cli): correctly determine emit state with redirects (#9287)
- fix(cli): early abort before type checking on missing modules (#9285)
- fix(cli): enable url wpt (#9299)
- fix(cli): fix panic in Deno.emit (#9302)
- fix(cli): fix panic in op_dns_resolve (#9187)
- fix(cli): fix recursive dispatches of unload event (#9207)
- fix(cli): fmt command help message (#9280)
- fix(cli): use DOMException in Performance#measure (#9142)
- fix(cli/flags): don't panic on invalid location scheme (#9202)
- fix(compile): fix panic when cross-compiling between windows and unix (#9203)
- fix(core): Handle prepareStackTrace() throws (#9211)
- fix(coverage): ignore comments (#8639)
- fix(coverage): use source maps when printing pretty reports (#9278)
- fix(lsp): complete list of unused diagnostics (#9274)
- fix(lsp): fix deadlocks, use one big mutex (#9271)
- fix(lsp): handle mbc documents properly (#9151)
- fix(lsp): handle mbc properly when formatting (#9273)
- fix(lsp): reduce deadlocks with in memory documents (#9259)
- fix(op_crates/fetch): fix ReadableStream.pipeThrough() (#9265)
- fix(op_crates/web): Add gb18030 and GBK encodings (#9242)
- fix(op_crates/web): Improve customInspect for Location (#9290)
- chore: new typescript WPT runner (#9269)

Changes in std version 0.85.0:

- feat(std/node): Add support for process.on("exit") (#8940)
- fix(std/async): make pooledMap() errors catchable (#9217)
- fix(std/node): Stop callbacks being called twice when callback throws error (#8867)
- fix(std/node): replace uses of `window` with `globalThis` (#9237)

1.7.0 / 2021.01.19

- BREAKING(unstable): Use hosts for net allowlists (#8845)
- BREAKING(unstable): remove CreateHttpClientOptions.caFile (#8928)
- feat(installer): Add support for MSYS on Windows (#8932)
- feat(unstable): add Deno.resolveDns API (#8790)
- feat(unstable): runtime compiler APIs consolidated to Deno.emit() (#8799, #9139)
- feat: Add WorkerOptions interface to type declarations (#9147)
- feat: Add configurable permissions for Workers (#8215)
- feat: Standalone lite binaries and cross compilation (#9141)
- feat: add --location= and globalThis.location (#7369)

- feat: add global tls session cache (#8877)
- feat: add markdown support to deno fmt (#8887)
- feat: add utf-16 and big5 to TextEncoder/TextDecoder (#8108)
- feat: denort binary (#9041)
- feat: stabilize Deno.shutdown() and Conn#closeWrite()(#9181)
- feat: support data urls (#8866)
- feat: support runtime flags for deno compile (#8738)
- feat: upload release zips to dl.deno.land (#9090)
- fix(cli): dispatch unload on exit (#9088)
- fix(cli): print a newline after help and version (#9158)
- fix(coverage): do not store source inline in raw reports (#9025)
- fix(coverage): merge duplicate reports (#8942)
- fix(coverage): report partial lines as uncovered (#9033)
- fix(inspector): kill child process after test (#8986)
- fix(install): escape % symbols in windows batch files (#9133)
- fix(install): fix cached-only flag (#9169)
- fix(lsp): Add textDocument/implementation (#9071)
- fix(lsp): Respect client capabilities for config and dynamic registration (#8865)
- fix(lsp): support specifying a tsconfig file (#8926)
- fix(op_crate/fetch): add back ReadableStream.getIterator and deprecate (#9146)
- fix(op_crate/fetch): align streams to spec (#9103)
- fix(op_crate/web): fix atob to throw spec aligned DOMException (#8798)
- fix(op_crates/fetch): correct regexp for fetch header (#8927)
- fix(op_crates/fetch): req streaming + 0-copy resp streaming (#9036)
- fix(op_crates/web) let TextEncoder#encodeInto accept detached ArrayBuffers (#9143)
- fix(op_crates/web): Use WorkerLocation for location in workers (#9084)
- fix(op_crates/websocket): respond to ping with pong (#8974)
- fix(watcher): keep working even when imported file has invalid syntax (#9091)
- fix: Use "none" instead of false to sandbox Workers (#9034)
- fix: Worker hangs when posting "undefined" as message (#8920)
- fix: align DOMException API to the spec and add web platform testing of it. (#9106)
- fix: don't error on version and help flag (#9064)
- fix: don't swallow customInspect exceptions (#9095)
- fix: enable WPT tests (#9072, #9087, #9013, #9016, #9047, #9012, #9007, #9004, #8990)
- fix: full commit hash in canary compile download (#9166)
- fix: ignore "use asm" (#9019)
- fix: implement DOMException#code (#9015)
- fix: incremental build for deno declaration files (#9138)
- fix: panic during `deno compile` with no args (#9167)
- fix: panic on invalid file:// module specifier (#8964)
- fix: race condition in file watcher (#9105)
- fix: redirect in --location relative fetch (#9150)
- fix: stronger input checking for setTimeout; add function overload (#8957)
- fix: use inline source maps when present in js (#8995)
- fix: use tokio for async fs ops (#9042)
- refactor(cli): remove 'js' module, simplify compiler snapshot (#9020)
- refactor(op_crates/crypto): Prefix ops with "op_crypto_" (#9067)
- refactor(op_crates/websocket): refactor event loop (#9079)
- refactor: Print cause chain when downcasting AnyError fails (#9059)
- refactor: make Process#kill() throw sensible errors on Windows (#9111)

- refactor: move WebSocket API to an op_crate (#9026)
- upgrade: Rust 1.49.0 (#8955)
- upgrade: deno_doc, deno_lint, dprint, swc_ecmascript, swc_bundler (#9003)
- upgrade: deno_lint to 0.2.16 (#9127)
- upgrade: rusty_v8 0.16.0, v8 8.9.255.3 (#9180)
- upgrade: swc_bundler 0.19.2 (#9085)
- upgrade: tokio 1.0 (#8779)

Changes in std version 0.84.0:

- BREAKING(std/wasi): make implementation details private (#8996)
- BREAKING(std/wasi): return exit code from start (#9022)
- feat(std/wasi): allow stdio resources to be specified (#8999)
- fix(std): Don't use JSDoc syntax for browser-compatibility headers (#8960)
- fix(std/http): Use ES private fields in server (#8981)
- fix(std/http): parsing of HTTP version header (#8902)
- fix(std/node): resolve files in symlinked directories (#8840)

1.6.3 / 2020.12.30
- feat(lsp): Implement textDocument/rename (#8910)
- feat(lsp): Add cache command (#8911)
- feat(unstable): collect coverage from the run command (#8893)
- fix: fetch bad URL will not panic (#8884)
- fix: info does not panic on missing modules (#8924)
- fix(core): Fix incorrect index in Promise.all error reporting (#8913)
- fix(lsp): handle ts debug errors better (#8914)
- fix(lsp): provide diagnostics for unresolved modules (#8872)
- upgrade: dprint, swc_bundler, swc_common, swc_ecmascript (#8901)
- upgrade: rusty_v8 0.15.0, v8 8.8.294 (#8898)

Changes in std version 0.83.0:

- feat(std/node): adds fs.mkdtemp & fs.mkdtempSync (#8604)
- fix(std/http): Don't expose ServerRequest::done as Deferred (#8919)

1.6.2 / 2020.12.22
- feat(lsp): support the unstable setting (#8851)
- feat(unstable): record raw coverage into a directory (#8642)
- feat(unstable): support in memory certificate data for Deno.createHttpClient (#8739)
- fix: atomically write files to $DENO_DIR (#8822)
- fix: implement ReadableStream fetch body handling (#8855)
- fix: make DNS resolution async (#8743)
- fix: make dynamic import errors catchable (#8750)
- fix: respect enable flag for requests in lsp (#8850)
- refactor: rename runtime/rt to runtime/js (#8806)
- refactor: rewrite lsp to be async (#8727)
- refactor: rewrite ops to use ResourceTable2 (#8512)
- refactor: optimise static assets in lsp (#8771)
- upgrade TypeScript to 4.1.3 (#8785)

Changes in std version 0.82.0:

- feat(std/node): Added os.type (#8591)

1.6.1 / 2020.12.14

- feat(lsp): support import maps (#8683)
- fix: show canary string in long version (#8675)
- fix: zsh completions (#8718)
- fix(compile): error when the output path already exists (#8681)
- fix(lsp): only resolve sources with supported schemas (#8696)
- fix(op_crates/fetch): support non-ascii response headers value (#8600)
- fix(repl): recover from invalid input (#8759)
- refactor: deno_runtime crate (#8640)
- upgrade: swc_ecmascript to 0.15.0 (#8688)

Changes in std version 0.81.0:

- fix(std/datetime): partsToDate (#8553)
- fix(std/wasi): disallow multiple starts (#8712)

1.6.0 / 2020.12.08

- BREAKING: Make "isolatedModules" setting non-configurable (#8482)
- feat: Add mvp language server (#8515, #8651)
- feat: deno compile (#8539, #8563, #8581)
- feat: Update to TypeScript 4.1 (#7573)
- feat: EventTarget signal support (#8616)
- feat: Add canary support to upgrade subcommand (#8476)
- feat(unstable): Add cbreak option to Deno.setRaw (#8383)
- fix: "onload" event order (#8376)
- fix: Add file URL support for Deno.readLink (#8423)
- fix: Add hygiene pass to transpile pipeline (#8586)
- fix: Require allow-write permissions for unixpackets datagrams & unix socket (#8511)
- fix: Highlight `async` and `of` in REPL (#8569)
- fix: Make output of deno info --json deterministic (#8483)
- fix: Panic in worker when closing at top level (#8510)
- fix: Support passing cli arguments under `deno eval` (#8547)
- fix: `redirect: "manual"` fetch should return `type: "default"` response (#8353)
- fix: close() calls sometimes prints results in REPL (#8558)
- fix: watcher doesn't exit when module resolution fails (#8521)
- fix: Fix PermissionDenied error being caught in Websocket constructor (#8402)
- fix: Set User-Agent header in Websocket (#8502, #8470)
- perf: Use minimal op with performance.now() (#8619)
- core: Implement new ResourceTable (#8273)
- core: Add FsModuleLoader that supports loading from filesystem (#8523)
- upgrade rusty_v8 to 0.14.0 (#8663)
- upgrade: deno_doc, deno_lint, dprint, swc (#8552, #8575, #8588)

Changes in std version 0.80.0:

- BREAKING(std/bytes): Adjust APIs based on std-wg discussion (#8612)
- feat(std/encoding/csv): Add stringify functionality (#8408)
- feat(std/fs): Re-enable `followSymlinks` on `walk()` (#8479)
- feat(std/http): Add Cookie value validation (#8471)
- feat(std/node): Add "setImmediate" and "clearImmediate" to global scope (#8566)
- feat(std/node): Port most of node errors (#7934)

- feat(std/node/stream): Add Duplex, Transform, Passthrough, pipeline, finished and promises (#7940)
- feat(std/wasi): Add return on exit option (#8605)
- feat(std/wasi): Add support for initializing reactors (#8603)
- feat(std/ws): protocol & version support (#8505)
- fix(std/bufio): Remove '\r' at the end of Windows lines (#8447)
- fix(std/encoding): Rewrite toml parser not to use eval() (#8624)
- fix(std/encoding/csv): Correct readme formatting due to dprint issues (#8503)
- fix(std/http): Prevent path traversal (#8474)
- fix(std/node): Inline default objects to ensure correct prototype (#8513)

1.5.4 / 2020.11.23

- feat(unstable): Add deno test --no-run (#8093)
- feat(unstable): Support --watch flag for bundle and fmt subcommands (#8276)
- fix: Support "deno run --v8-flags=--help" without script (#8110)
- fix(tsc): Allow non-standard extensions on imports (#8464)
- refactor: Improve Deno.version type declaration (#8391)
- refactor: Rename --failfast to --fail-fast for test subcommand (#8456)
- upgrade: rusty_v8 0.13.0, v8 8.8.278.2 (#8446)

Changes in std version 0.79.0:

- feat(std/hash): Add HmacSha1 (#8418)
- feat(std/http): Check if cookie property is valid (#7189)
- feat(std/http): Validate cookie path value (#8457)
- feat(std/io): ReadableStream from AsyncIterator & WritableStream from Writer (#8378)
- feat(std/log): Log error stack (#8401)
- feat(std/node): Add os.totalmem, os.freemem (#8317)
- feat(std/node): Add ReadableStream, WritableStream, errors support (#7569)
- feat(std/node): Add util.deprecate (#8407)
- feat(std/node): Add process.nextTick (#8386)
- fix(std/http): Fix error handling in the request iterator (#8365)
- fix(std/node) Fix event extendability (#8409)
- fix(std/node): Correct typings for global, globalThis, window (#8363)

1.5.3 / 2020.11.16

- feat(unstable): support deno lint --rules --json (#8384)
- fix: fix various global objects constructor length (#8373)
- fix: allow declaration emits for Deno.compile() (#8303)
- fix: allow root modules be .mjs/.cjs (#8310)
- fix: allow setting of importsNotUsedAsValues in Deno.compile() (#8306)
- fix: do not write tsbuildinfo when diagnostics are emitted (#8311)
- fix: don't walk the subdirectory twice when using the `--ignore` flag (#8040, #8375)
- fix: local sources are not cached in memory (#8328)
- fix: Use safe shell escaping in `deno install` (#7613)
- fix: DOM handler order in Websocket and Worker (#8320, #8334)
- fix(op_crates/web) make isTrusted not constructable (#8337)
- fix(op_crates/web): FileReader event handler order (#8348)
- fix(op_crates/web): handler order when reassign (#8264)
- refactor: deno_crypto op crate (#7956)

Changes in std version 0.78.0:

- feat(std/node): consistent Node.js builtin shapes (#8274)
- fix(std/http): flush body chunks for HTTP chunked encoding (#8349)
- refactor(std/fs): moved isCopyFolder to options (#8319)

1.5.2 / 2020.11.09

- fix(core/error): Remove extra newline from JsError::fmt() (#8145)
- fix(op_crates/web): make TextEncoder work with forced non-strings (#8206)
- fix(op_crates/web): fix URLSearchParams, malformed url handling (#8092)
- fix(op_crates/web): define abort event handler on prototype (#8230)
- fix(cli/repl): Fixing syntax highlighting (#8202)
- fix: inject helpers when transpiling via swc (#8221)
- fix: add commit hash and target to long_version output (#8133)
- fix: correct libs sent to tsc for unstable worker (#8260)
- fix: properly handle type checking root modules with type definitions (#8263)
- fix: allow remapping to locals for import map (#8262)
- fix: ensure that transitory dependencies are emitted (#8275)
- fix: make onabort event handler web compatible (#8225)
- fix: display of non-ASCII characters on Windows (#8199)
- refactor: Cleanup Flags to Permissions conversion (#8213)
- refactor: migrate runtime compile/bundle to new infrastructure (#8192)
- refactor: cleanup compiler snapshot and tsc/module_graph (#8220)
- refactor: remove ProgramState::permissions (#8228)
- refactor: refactor file_fetcher (#8245)
- refactor: rewrite permission_test to not depend on Python (#8291)
- refactor: auto detect target triples for upgrade (#8286)
- build: migrate to dlint (#8176)
- build: remove eslint (#8232)
- build: rewrite tools/ scripts to deno (#8247)
- build: full color ci logs (#8280)
- upgrade: TypeScript to 4.0.5 (#8138)
- upgrade: deno_doc, deno_lint, dprint, swc (#8292)

Changes in std version 0.77.0:

- feat(std/node/fs): add realpath and realpathSync (#8169)
- feat(std/wasi): add start method to Context (#8141)
- fix(std/flags): Fix parse incorrectly parsing alias flags with equals (#8216)
- fix(std/node): only define Node.js globals when loading std/node/global (#8281)

1.5.1 / 2020.10.31

- fix: Accept Windows line breaks in prompt/confirm/alert (#8149)
- fix: Deno.fdata(), Deno.fdatasync() added to stable (#8193)
- fix: Strip "\?" prefix when displaying Windows paths (#8135)
- fix: Make hashes of tsconfig deterministic (#8167)
- fix: Module graph handles redirects properly (#8159)
- fix: Restore tripleslash lib refs support (#8157)
- fix: Panic in bundler (#8168)
- fix(repl): Don't hang on unpaired braces (#8151)
- refactor: Don't spin up V8 for `deno cache` (#8186)
- refactor: Create a single watcher for whole process (#8083)
- upgrade: deno_doc, deno_lint, dprint, swc (#8197)

Changes in std version 0.76.0:

- feat(std/node/crypto): Add randomBytes and pbkdf2 (#8191)
- fix(std/wasi): Remove stray console.log call (#8156)

1.5.0 / 2020.10.27

- BREAKING: Enable isolatedModules by default (#8050)
- feat(bundle): Add support for --no-check (#8023)
- feat(console): Inspect with colors regardless of Deno.noColor (#7778)
- feat(doc): Support --import-map flag (#7821)
- feat(fmt): Make --ignore flag stable (#7922)
- feat(install): Add missing flags for deno install (#7601)
- feat(repl): Add regex based syntax highlighter (#7811)
- feat(repl): Add tab completion (#7827)
- feat(test): Pass script args to test command (#8121)
- feat(unstable): Add Deno.sleepSync() (#7974)
- feat(unstable): Add Deno.systemCpuInfo() (#7774)
- feat: Add alert, confirm, and prompt (#7507)
- feat: Add types for WeakRef/FinalizationRegistry (#8056)
- feat: Stabilize Deno.fsync and Deno.fdatasync (#8038)
- fix(console): Fix the test cases of function inspections (#7965)
- fix(console): Only inspect getters with option (#7830)
- fix(core): Indicate exceptions in promises (#8124)
- fix(core): Top Level Await module execution (#7946)
- fix(op_crates/fetch): Body.body should be stream of Uint8Array (#8030)
- fix(op_crates/fetch): Ensure Request.method is a string (#8100)
- fix(op_crates/web): Better TextEncoder error message (#8005)
- fix(op_crates/web): Expose event properties in console output (#8103)
- fix(op_crates/web): TextEncoder should throw RangeError (#8039)
- fix(op_crates/web): URL.pathname backslash replacement (#7937)
- fix(repl): Ignore pair matching inside literals (#8037)
- fix(repl): Keyboard interrupt should continue (#7960)
- fix(repl): Unterminated string literal should invalidate (#7896)
- fix(repl): Write all results to stdout (#7893)
- fix(rt/main): Add global interface objects (#7875)
- fix(rt/performance): Check for object props in startOrMeasureOptions (#7884)
- fix(rt/websockets): Only add Sec-WebSocket-Protocol if not empty (#7936)
- fix(test): Return error when awaiting unresolved promise (#7968)
- fix: Do not throw on empty typescript files (#8143)
- fix: Fix inspection of Function (#7930)
- fix: Handle URL paths in Deno.mkdir() (#8140)
- fix: Handling of relative importmaps while using watch (#7950)
- fix: Print error stacks from the origin Worker (#7987)
- fix: Restore permission check on workers (#8123)
- fix: Use -rw-r--r-- for cache files (#8132)
- fix: Use rid getter for stdio (#8014)
- fix: handle roots with extensions that don't match media type (#8114)
- refactor(core): more control over isolate creation (#8000)
- refactor: New TSC infrastructure (#7996, #7981, #7892)
- refactor: Rename --importmap to --import-map (#7032)
- refactor: Rewrite Deno.transpileOnly() to use SWC (#8090)

- upgrade: deno_doc, deno_lint, dprint, swc (#8009, #8077)
- upgrade: rusty_v8 and v8 8.7.220.3 (#8017)

Changes in std version 0.75.0:

- feat(std/fs/node): Add more APIs (#7921)
- feat(std/path): Add toFileUrl() (#7971)
- feat(std/testing): Add assertExists assertion (#7874)
- feat(std/testing): Add assertObjectMatch assertion (#8001)
- fix(std/http): Path traversal in file_server.ts (#8134)
- fix(std/toml): Parsing inline arrays of inline tables (#7902)
- fix(std/encoding): base64 properly encodes mbc and handles Uint8Arrays (#7807)
- fix(std/http/file_server): File server should ignore query params (#8116)
- fix(std/node): Buffer.copy doesn't work as expected (#8125)
- fix(std/wasi): Disallow path_open outside of pre-opened dirfd (#8078)
- refactor(std/testing): Rename assert_Contains to assert_Includes (#7951)

1.4.6 / 2020.10.10

- fix: 100% CPU idling problem by reverting #7672 (#7911)
- fix(op_crate/web): add padding on URLSearchParam (#7905)
- fix(op_crates/fetch): Stringify and parse Request URLs (#7838)
- refactor(core): Implement Serialize for ModuleSpecifier (#7900)
- upgrade: Rust 1.47.0 (#7886)

1.4.5 / 2020.10.08

- feat(unstable): Revert "enable importsNotUsedAsValues by default #7413" (#7800)
- fix: Update worker types to better align to lib.dom.d.ts (#7843)
- fix(cli/ops/fs): Preserve Windows path separators in Deno.realPath() (#7833)
- fix(cli/rt/console): Don't require a prototype to detect a class instance (#7869)
- fix(cli/rt/error_stack): Improve message line formatting (#7860)
- fix(core): Handle unregistered errors in core better (#7817)
- fix(core): Module execution with top level await (#7672)
- perf(cli/console): Don't add redundant ANSI codes (#7823)
- refactor(cli): Remove TextDocument (#7850)
- refactor(cli/inspector): Use &str for post_message (#7851)
- refactor(cli/repl): Tightly integrate event loop (#7834)
- refactor(core): Cleanup JsRuntime (#7853, #7855, #7825, #7846)
- upgrade: deno_doc, deno_lint, dprint, swc (#7862)
- upgrade: rusty_v8 0.11.0, V8 8.7.220.3 (#7859)

Changes in std version 0.74.0:

- chore(std/http): Rename http_bench.ts -> bench.ts (#7509)
- feat(std/node/fs): Adding readdir, rename, and some others (#7666)
- fix(std/node/fs): Allow appendFileSync to accept Uint8Array as type for data (#7835)

1.4.4 / 2020.10.03

- fix(cli): Update type definitions to align to TS dom (#7791)
- fix(cli/repl): Fix hot loop in REPL (#7804)
- fix(cli/repl): Enable colors on inspected values (#7798)

1.4.3 / 2020.10.02

- feat(unstable): Add module specifier to deno info --json output (#7725)
- fix: Bundle loader returns exported value (#7764)
- fix: Check cached versions during transpile (#7760)
- fix: Net listen crashes on explicit undefined hostname (#7706)
- fix: --no-check recognizes require (#7720)
- fix: Use $deno$test.ts instead of .deno.test.ts (#7717)
- fix: Use global_state file_fetcher when using SpecifierHandler (#7748)
- fix(console): Catch and format getter errors (#7766)
- fix(dts): Use var instead of const and let for globals (#7680)
- fix(inspector): Shutdown server gracefully on drop (#7716)
- fix(repl): Enable await and let re-declarations (#7784)
- fix(repl): Use a default referrer when empty (#7794)
- fix(test): Do not start inspector server when collecting coverage (#7718)
- fix(websocket): Add missing close events and remove extra error event (#7606)
- refactor: Add concept of 'legacy' compiler to enable non-breaking refactoring (#7762)
- refactor: Combine MainWorker::new and MainWorker::create (#7693)
- refactor: Extract inspector session (#7756, #7763)
- refactor: Factor out check_unstable op helper (#7695)
- refactor: Improve graph and tsc_config (#7747)
- refactor: Improve op crate interfaces for other consumers (#7745)
- refactor: Move op state registration to workers (#7696)
- refactor: Use JsRuntime to implement TSC (#7691)
- refactor: Add Deno.InspectOptions::colors (#7742)
- upgrade: swc, deno_doc, deno_lint, dprint (#7711, #7793)

Changes in std version 0.72.0:

- BREAKING(std/encoding/csv): Improve the definition of ParseOptions (#7714)
- feat(std/path): Align globToRegExp() with bash glob expansion (#7209)
- fix(std/datetime): Add timezone to date strings in tests (#7675)
- refactor(std/example): Inconsistencies in the example tests (#7684)
- refactor(std/testing): Get rid of default export and make std/testing/diff.ts private (#7592)

1.4.2 / 2020.09.25

- fix: Better formatting in console (#7642, #7641, #7553)
- fix: Change log level to which prefix added (#7582)
- fix: Change the Console class declaration to an interface (#7646)
- fix: Clearing timers race condition (#7617)
- fix: customInspect works on functions (#7670)
- fix: Ignore fileExists in tsc host (#7635)
- fix: Make --unstable a global flag (#7585)
- fix: Make --watch and --inspect conflicting args (#7610)
- fix: Make some web API constructors illegal at runtime (#7468)
- fix: Replaced legacy chrome-devtools:// scheme. (#7659)
- fix: Response.arrayBuffer() doesn't return promise (#7618)
- fix: Update supported text encodings (#7668)
- fix: Use class instead of var+interface in d.ts #7514
- fix(coverage): print lines with no coverage to stdout (#7640)
- fix(fmt,lint): do not print number of checked files when `--quiet` is enabled (#7579)
- fix(info): add --importmap flag (#7424)
- fix(installer): Don't reload by default (#7596)

- fix(repl): interpret object literals as expressions (#7591)
- fix(watch): watch importmap file for changes (#7580)
- refactor(core): support error stack, remove js_check (#7629, #7636)
- refactor(coverage): Harden coverage collection (#7584, #7616, #7577)
- upgrade: TypeScript to 4.0.3 (#7637)
- example(core): Add hello world example (#7611)

Changes in std version 0.71.0:

- feat(std/node): implement getSystemErrorName() (#7624)
- fix(std/datetime): 12 and 24 support (#7661)
- fix(std/fs): mark createWalkEntry(Sync) as internal (#7643)
- chore(std/hash): update crates (#7631)

1.4.1 / 2020.09.18

- fix(cli/console): escape special characters in strings and property names (#7546, #7533, #7550)
- fix(cli/fmt): canonicalize files in current dir (#7508)
- fix(cli/fmt): make fmt output more readable (#7534)
- fix(cli/install): revert "bundle before installation" (#7522)
- fix(cli/js): disable URL.createObjectUrl (#7543)
- fix(cli/js): use Buffer.writeSync in MultipartBuilder (#7542)
- fix(cli/repl): disable rustyline logs (#7535)
- fix(cli/repl): format evaluation results with the object specifier (#7561)
- fix(cli/bundle,eval,repl): add missing flags (#7414)
- refactor(cli): move fetch() implementation to op_crates/fetch (#7524, #7529)
- refactor(cli): move FileReader and URL to op_crates/web (#7554, #7544)
- refactor(cli): move op_resources and op_close to deno_core (#7539)
- refactor(cli/info,unstable): deno info --json output (#7417)
- refactor(cli/js): simplify global properties (#7502)
- refactor(cli/js): use Symbol.for instead of Symbol (#7537)
- refactor(core): remove JsRuntime::set_js_error_create_fn (#7478)
- refactor(core): use the 'anyhow' crate instead of ErrBox (#7476)
- upgrade: rust crates (#7454)
- benchmark: add no_check_hello benchmark (#7458)

Changes in std version 0.70.0:

- feat(std/node): add AssertionError class (#7210)
- fix(std/datetime): timezone bug (#7466)
- fix(std/testing): assertion diff color (#7499)

1.4.0 / 2020.09.13

- feat: Implement WebSocket API (#7051, 7437)
- feat(console): print proxy details (#7139)
- feat(console): support CSS styling with "%c" (#7357)
- feat(core): Add JSON ops (#7336)
- feat(fmt, lint): show number of checked files (#7312)
- feat(info): Dependency count and sizes (#6786, #7439)
- feat(install): bundle before installation (#5276)
- feat(op_crates/web): Add all single byte encodings to TextDecoder (#6178)
- feat(unstable): Add Deno.systemMemoryInfo() (#7350)
- feat(unstable): deno run --watch (#7382)

- feat(unstable): deno test --coverage (#6901)
- feat(unstable): enable importsNotUsedAsValues by default (#7413)
- feat(unstable): enable isolatedModules by default (#7327)
- fix: Empty Response body returns 0-byte array (#7387)
- fix: panic on process.kill() after run (#7405)
- fix: colors mismatch (#7367)
- fix: compiler config resolution using relative paths (#7392)
- fix(core): panic on big string allocation (#7395)
- fix(op_crates/web): Use "deno:" URLs for internal script specifiers (#7383)
- refactor: Improve placeholder module names (#7430)
- refactor: improve tsc diagnostics (#7420)
- refactor(core): merge CoreIsolate and EsIsolate into JsRuntime (#7370, #7373, #7415)
- refactor(core): Use gotham-like state for ops (#7385)
- upgrade: deno_doc, deno_lint, dprint, swc (#7381, #7391, #7402, #7434)
- upgrade: rusty_v8 0.10.0 / V8 8.7.75 (#7429)

Changes in std version 0.69.0:

- BREAKING(std/fs): remove writeJson and writeJsonSync (#7256)
- BREAKING(std/fs): remove readJson and readJsonSync (#7255)
- BREAKING(std/ws): remove connect method (#7403)

1.3.3 / 2020.09.04

- feat(unstable): Add Deno.futime and Deno.futimeSync (#7266)
- feat(unstable): Allow deno lint to read from stdin (#7263)
- fix: Don't expose globalThis.__bootstrap (#7344)
- fix: Handle bad redirects more gracefully (#7342)
- fix: Handling of + character in URLSearchParams (#7314)
- fix: Regex for TS references and deno-types (#7333)
- fix: Set maximum size of thread pool to 31 (#7290)
- fix: Support missing features in --no-check (#7289)
- fix: Use millisecond precision for Deno.futime and Deno.utime (#7299)
- fix: Use upstream type definitions for WebAssembly (#7216)
- refactor: Compiler config in Rust (#7228)
- refactor: Support env_logger / RUST_LOG (#7142)
- refactor: Support multiline diagnostics in linter (#7303)
- refactor: Use dependency analyzer from SWC (#7334)
- upgrade: rust 1.46.0 (#7251)
- upgrade: swc, deno_doc, deno_lint, dprint (#7276, #7332)

Changes in std version 0.68.0:

- refactor(std/uuid): remove dependency on isString from std/node (#7273)

1.3.2 / 2020.08.29

- fix(cli): revert "never type check deno info #6978" (#7199)
- fix(console): handle escape sequences when logging objects (#7171)
- fix(doc): stack overflow for .d.ts files (#7167)
- fix(install): Strip "@..." suffixes from inferred names (#7223)
- fix(lint): use recommended rules set (#7222)
- fix(url): Add missing part assignment (#7239)
- fix(url): Don't encode "'" in non-special query strings (#7152)

- fix(web): throw TypeError on invalid input types in TextDecoder.decode() (#7179)
- build: Move benchmarks to Rust (#7134)
- upgrade: swc, dprint, deno_lint, deno_doc (#7162, #7194)
- upgrade: rusty_v8 0.9.1 / V8 8.6.334 (#7243)
- upgrade: TypeScript 4.0 (#6514)

Changes in std version 0.67.0:

- BREAKING(std/wasi): rename Module to Context (#7110)
- BREAKING(std/wasi): use record for exports (#7109)
- feat(std/fmt): add bright color variations (#7241)
- feat(std/node): add URL export (#7132)
- feat(std/testing): add assertNotMatch (#6775)
- fix(std/encoding/toml): Comment after arrays causing incorrect output (#7224)
- fix(std/node): "events" and "util" modules (#7170)
- fix(std/testing): invalid dates assertion equality (#7230)
- fix(std/wasi): always capture syscall exceptions (#7116)
- fix(std/wasi): ignore lint errors (#7197)
- fix(std/wasi): invalid number to bigint conversion in fd_tell (#7215)
- fix(std/wasi): return flags from fd_fdstat_get (#7112)

1.3.1 / 2020.08.21

- fix: Allow isolated "%"s when parsing file URLs (#7108)
- fix: Blob.arrayBuffer returns Uint8Array (#7086)
- fix: CLI argument parsing with dash values (#7039)
- fix: Create Body stream from any valid bodySource (#7128)
- fix: Granular permission requests/revokes (#7074)
- fix: Handling of multiple spaces in URLSearchParams (#7068)
- core: Enable WebAssembly.instantiateStreaming (#7043)
- core: Add missing export of HeapLimits (#7047)
- upgrade: swc_ecmascript, deno_lint, dprint (#7098)

Changes in std version 0.66.0:

- BREAKING(std/datetime): Remove currentDayOfYear (#7059)
- feat(std/node): Add basic asserts (#7091)
- feat(std/datetime): Generalise parser, add formatter (#6619)
- fix(std/node): Misnamed assert exports (#7123)
- fix(std/encoding/toml): Stop TOML parser from detecting numbers in strings. (#7064)
- fix(std/encoding/csv): Improve error message on ParseError (#7057)

1.3.0 / 2020.08.13

Changes in the CLI:

- feat: Add "--no-check" flag to deno install (#6948)
- feat: Add "--ignore" flag to deno lint (#6934)
- feat: Add "--json" flag to deno lint (#6940)
- feat: Add "--reload" flag to deno bundle (#6996)
- feat: Add "--reload" flag to deno info (#7009)
- feat: FileReader API (#6673)
- feat: Handle imports in deno doc (#6987)
- feat: Stabilize Deno.mainModule (#6993)

- feat: Support file URLs in Deno.run for executable (#6994)
- fix: console.log should see color codes when grouping occurs (#7000)
- fix: URLSearchParams.toString() behaviour is different from browsers (#7017)
- fix: Remove @ts-expect-error directives (#7024)
- fix(unstable): Add missing globals to diagnostics (#6988)
- refactor(doc): Remove detailed / summary distinction (#6818)
- core: Memory limits & callbacks (#6914)
- upgrade: TypeScript to 3.9.7 (#7036)
- upgrade: Rust crates (#7034, #7040)

Changes in std version 0.65.0:

- feat(std/http): Add TLS serve abilities to file_server (#6962)
- feat(std/http): Add --no-dir-listing flag to file_server (#6808)
- feat(std/node): Add util.inspect (#6833)
- fix: Make std work with isolatedModules (#7016)

1.2.3 / 2020.08.08

Changes in the CLI:

- fix: Never type check in deno info (#6978)
- fix: add missing globals to unstable diagnostics (#6960)
- fix: add support for non-UTF8 source files (#6789)
- fix: hash file names in gen cache (#6911)
- refactor: Encode op errors as strings instead of numbers (#6977)
- refactor: Op crate for Web APIs (#6906)
- refactor: remove repeated code in main.rs (#6954)
- upgrade to rusty_v8 0.8.1 / V8 8.6.334 (#6980)
- upgrade: deno_lint v0.1.21 (#6985)
- upgrade: swc_ecmascript (#6943)
- feat(unstable): custom http client for fetch (#6918)

Changes in std version 0.64.0:

- fix(std/toml): parser error with inline comments (#6942)
- fix(std/encoding/toml): Add boolean support to stringify (#6941)
- refactor: Rewrite globToRegExp() (#6963)

1.2.2 / 2020.07.31

Changes in the CLI:

- fix: Change release build flags to optimize for size (#6907)
- fix: Fix file URL to path conversion on Windows (#6920)
- fix: deno-types, X-TypeScript-Types precedence (#6761)
- fix: downcast from SwcDiagnosticBuffer to OpError (#6909)
- perf: Use SWC to strip types for "--no-check" flag (#6895)
- upgrade: deno_lint, dprint, swc (#6928, #6869)
- feat(unstable): add "--ignore" flag to deno fmt (#6890)

Changes in std version 0.63.0:

- feat(std/async): add pooledMap utility (#6898)
- fix(std/json): Add newline at the end of json files (#6885)
- fix(std/path): Percent-decode in fromFileUrl() (#6913)

- fix(std/tar): directory type bug (#6905)

1.2.1 / 2020.07.23

Changes in the CLI:

- fix: IPv6 hostname should be compressed (#6772)
- fix: Ignore polling errors caused by return() in watchFs (#6785)
- fix: Improve URL compatibility (#6807)
- fix: ModuleSpecifier removes relative path parts (#6762)
- fix: Share reqwest client between fetch calls (#6792)
- fix: add icon and metadata to deno.exe on Windows (#6693)
- fix: panic for runtime error in TS compiler (#6758)
- fix: providing empty source code for missing compiled files (#6760)
- refactor: Make OpDispatcher a trait (#6736, #6742)
- refactor: Remove duplicate code and allow filename overwrite for DomFile (#6817, #6830)
- upgrade: Rust 1.45.0 (#6791)
- upgrade: rusty_v8 0.7.0 (#6801)
- upgrade: tokio 0.2.22 (#6838)

Changes in std version 0.62.0:

- BREAKING(std/fs): remove readFileStr and writeFileStr (#6848, #6847)
- feat(std/encoding): add ascii85 module (#6711)
- feat(std/node): add string_decoder (#6638)
- fix(std/encoding/toml): could not parse strings with apostrophes/semicolons (#6781)
- fix(std/testing): assertThrows inheritance (#6623)
- fix(std/wasi): remove number overload from rights in path_open (#6768)
- refactor(std/datetime): improve weekOfYear (#6741)
- refactor(std/path): enrich the types in parse_format_test (#6803)

1.2.0 / 2020.07.13

Changes in the CLI:

- feat(cli): Add --cert option to "deno upgrade" (#6609)
- feat(cli): Add --config flag to "deno install" (#6204)
- feat(cli): Add --json option to "deno info" (#6372)
- feat(cli): Add --no-check option (#6456)
- feat(cli): Add --output option to "deno upgrade" (#6352)
- feat(cli): Add DENO_CERT environment variable (#6370)
- feat(cli): Add lockfile support to bundle (#6624)
- feat(cli/js): Add WriteFileOptions to writeTextFile & writeTextFileSync (#6280)
- feat(cli/js): Add copy argument to Buffer.bytes (#6697)
- feat(cli/js): Add performance user timing APIs (#6421)
- feat(cli/js): Add sorted, trailingComma, compact and iterableLimit to InspectOptions (#6591)
- feat(cli/js): Deno.chown() make uid, gid args optional (#4612)
- feat(doc): Improve terminal printer (#6594)
- feat(test): Add support for regex in filter flag (#6343)
- feat(unstable): Add Deno.consoleSize() (#6520)
- feat(unstable): Add Deno.ppid (#6539, #6717)
- fix(cli): Don't panic when no "HOME" env var is set (#6728)
- fix(cli): Harden pragma and reference parsing in module analysis (#6702)
- fix(cli): Panic when stdio is null on windows (#6528)

- fix(cli): Parsing of --allow-net flag (#6698)
- fix(cli/js): Allow Buffer to store MAX_SIZE bytes (#6570)
- fix(cli/js): Definition of URL constructor (#6653)
- fix(cli/js): Deno.setRaw shouldn't panic on ENOTTY (#6630)
- fix(cli/js): Fix process socket types (#6676)
- fix(cli/js): Fix relative redirect in fetch API (#6715)
- fix(cli/js): Implement IPv4 hostname parsing in URL (#6707)
- fix(cli/js): Implement spec-compliant host parsing for URL (#6689)
- fix(cli/js): Response constructor default properties in fetch API (#6650)
- fix(cli/js): Update timers to ignore Date Override (#6552)
- perf(cli): Improve .arrayBuffer() speed in fetch API (#6669)
- refactor(core): Remove control slice from ops (#6048)

Changes in std version 0.61.0:

- BREAKING(std/encoding/hex): Simplify API (#6690)
- feat(std/datetime): Add weekOfYear (#6659)
- feat(std/log): Expose Logger type and improve public interface for get & set log levels (#6617)
- feat(std/node): Add buf.equals() (#6640)
- feat(std/wasi): Implement fd_readdir (#6631)
- fix(std): base64 in workers (#6681)
- fix(std): md5 in workers (#6662)
- fix(std/http): Properly return port 80 in _parseAddrFromStr (#6635)
- fix(std/mime): Boundary random hex values (#6646)
- fix(std/node): Add encoding argument to Buffer.byteLength (#6639)
- fix(std/testing/asserts): AssertEquals/NotEquals should use milliseconds in Date (#6644)
- fix(std/wasi): Return errno::success from fd_tell (#6636)

1.1.3 / 2020.07.03

Changes in the CLI:

- fix(cli): Change seek offset type from i32 to i64 (#6518)
- fix(cli/body): Maximum call stack size exceeded error (#6537)
- fix(cli/doc): Doc printer missing [] around tuple type (#6523)
- fix(cli/js): Buffer.bytes() ArrayBuffer size (#6511)
- fix(cli/js): Fix conditional types for process sockets (#6275)
- fix(cli/upgrade): Upgrade fails on Windows with space in temp path (#6522)
- fix: Lock file for dynamic imports (#6569)
- fix: Move ImportMeta to deno.ns lib (#6588)
- fix: Net permissions didn't account for default ports (#6606)
- refactor: Improvements to TsCompiler and its tests (#6576)
- upgrade: deno_lint 0.1.15 (#6580, #6614)
- upgrade: dprint-plugin-typescript 0.19.5 (#6527, #6614)

Changes in std version 0.60.0:

- feat(std/asserts): Allow assert functions to specify type parameter (#6413)
- feat(std/datetime): Add is leap and difference functions (#4857)
- feat(std/io): Add fromStreamReader, fromStreamWriter (#5789, #6535)
- feat(std/node): Add Buffer.allocUnsafe (#6533)
- feat(std/node): Add Buffer.isEncoding (#6521)
- feat(std/node): Support hex/base64 encoding in fs.readFile/fs.writeFile (#6512)

- feat(std/wasi) Implement fd_filestat_get (#6555)
- feat(std/wasi) Implement fd_filestat_set_size (#6558)
- feat(std/wasi): Implement fd_datasync (#6556)
- feat(std/wasi): Implement fd_sync (#6560)
- fix(std/http): Catch errors on file_server response.send (#6285)
- fix(std/http): Support ipv6 parsing (#5263)
- fix(std/log): Print "{msg}" when log an empty line (#6381)
- fix(std/node): Add fill & encoding args to Buffer.alloc (#6526)
- fix(std/node): Do not use absolute urls (#6562)
- fix(std/wasi): path_filestat_get padding (#6509)
- fix(std/wasi): Use lookupflags for path_filestat_get (#6530)
- refactor(std/http): Cookie types to not require full ServerRequest object (#6577)

1.1.2 / 2020.06.26

Changes in the CLI:

- fix(web/console): Improve string quoting behaviour (#6457)
- fix(web/url): Support UNC paths on Windows (#6418)
- fix(web/url): Support URLSearchParam as Body (#6416)
- fix: 'Compile' messages changed to 'Check' messages (#6504)
- fix: Panic when process stdio rid is 0 or invalid (#6405)
- fix: enable experimental-wasm-bigint (#6443)
- fix: ipv6 parsing for --allow-net params (#6453, #6472)
- fix: panic when demanding permissions for hostless URLs (#6500)
- fix: strings shouldn't be interpreted as file URLs (#6412)
- refactor: Add ability to output compiler performance information (#6434)
- refactor: Incremental compilation for TypeScript (#6428, #6489)
- upgrade: rusty_v8 0.4.2 / V8 8.5.216 (#6503)

Changes in unstable APIs:

- Add Deno.fdatasyncSync and fdatasync (#6403)
- Add Deno.fstatSync and fstat (#6425)
- Add Deno.fsyncSync and fsync (#6411)
- Add Deno.ftruncate and ftruncateSync (#6243)
- Remove Deno.dir (#6385)

Changes in std version 0.59.0:

- BREAKING(std/encoding/hex): reorder encode & decode arguments (#6410)
- feat(std/node): support hex / base64 encoding in Buffer (#6414)
- feat(std/wasi): add wasi_snapshot_preview1 (#6441)
- fix(std/io): Make BufWriter/BufWriterSync.flush write all chunks (#6269)
- fix(std/node): fix readFile types, add encoding types (#6451)
- fix(std/node): global process should usable (#6392)
- fix(std/node/process): env, argv exports (#6455)
- fix(std/testing) assertArrayContains should work with any array-like (#6402)
- fix(std/testing): assertThrows gracefully fails if non-Error thrown (#6330)
- refactor(std/testing): Remove unuseful statement (#6486)
- refactor: shift copyBytes and tweak deps to reduce dependencies (#6469)

1.1.1 / 2020.06.19

- fix: "deno test" should respect NO_COLOR=true (#6371)
- fix: Deno.bundle supports targets < ES2017 (#6346)
- fix: decode path properly on win32 (#6351)
- fix: improve failure message for deno upgrade (#6348)
- fix: apply http redirection limit for cached files (#6308)
- fix: JSX compilation bug and provide better error message (#6300)
- fix: DatagramConn.send (unstable) should return bytes sent (#6265, #6291)
- upgrade: v8 to 8.5.104, rusty_v8 0.5.1 (#6377)
- upgrade: crates (#6378)

Changes in std version 0.58.0:

- feat(std/log): expose logger name to LogRecord (#6316)
- fix(std/async): MuxAsyncIterator throws muxed errors (#6295)
- fix(std/io): BufWriter/StringWriter bug (#6247)
- fix(std/io): Use Deno.test in writers_test (#6273)
- fix(std/node): added tests for static methods of Buffer (#6276)
- fix(std/testing): assertEqual so that it handles URL objects (#6278)
- perf(std/hash): reimplement all hashes in WASM (#6292)

1.1.0 / 2020.06.12

Changes in the CLI:

- feat: "deno eval -p" (#5682)
- feat: "deno lint" subcommand (#6125, #6208, #6222, #6248, #6258, #6264)
- feat: Add Deno.mainModule (#6180)
- feat: Add Deno.env.delete() (#5859)
- feat: Add TestDefinition::only (#5793)
- feat: Allow reading the entry file from stdin (#6130)
- feat: Handle .mjs files in "deno test" and "deno fmt" (#6134, #6122)
- feat: URL support in Deno filesystem methods (#5990)
- feat: make rid on Deno.Listener public (#5571)
- feat(core): Add unregister op (#6214)
- feat(doc): Display all overloads in cli details view (#6186)
- feat(doc): Handle detail output for enum (#6078)
- feat(fmt): Add diff for "deno fmt --check" (#5599)
- fix: Handle @deno-types in export {} (#6202)
- fix: Several regressions in TS compiler (#6177)
- fix(cli): 'deno upgrade' doesn't work on Windows 8.1/PowerShell 4.0 (#6132)
- fix(cli): WebAssembly runtime error propagation (#6137)
- fix(cli/js/buffer): Remove try-catch from Buffer.readFrom, readFromSync (#6161)
- fix(cli/js/io): Deno.readSync on stdin (#6126)
- fix(cli/js/net): UDP BorrowMutError (#6221)
- fix(cli/js/process): Always return a code in ProcessStatus (#5244)
- fix(cli/js/process): Strengthen socket types based on pipes (#4836)
- fix(cli/js/web): IPv6 hostname support in URL (#5766)
- fix(cli/js/web/worker): Disable relative module specifiers (#5266)
- fix(cli/web/fetch): multipart/form-data request body support for binary files (#5886)
- fix(core): ES module snapshots (#6111)
- revert: "feat: format deno bundle output (#5139)" (#6085)
- upgrade: Rust 1.44.0 (#6113)
- upgrade: swc_ecma_parser 0.24.5 (#6077)

Changes in std version 0.57.0:

- feat(std/encoding/binary): Add varnumBytes(), varbigBytes() (#5173)
- feat(std/hash): Add sha3 (#5558)
- feat(std/log): Inline and deferred statement resolution logging (#5192)
- feat(std/node): Add util.promisify (#5540)
- feat(std/node): Add util.types (#6159)
- feat(std/node): Buffer (#5925)
- feat(std/testing): Allow non-void promises in assertThrowsAsync (#6052)
- fix(http/server): Flaky test on Windows (#6188)
- fix(std/archive): Untar (#6217) cleanup std/tar (#6185)
- fix(std/http): Don't use assert() for user input validation (#6092)
- fix(std/http): Prevent crash on UnexpectedEof and InvalidData (#6155)
- fix(std/http/file_server): Args handling only if invoked directly (#5989)
- fix(std/io): StringReader implementation (#6148)
- fix(std/log): Revert setInterval log flushing as it prevents process completion (#6127)
- fix(std/node): Emitter.removeAllListeners (#5583)
- fix(std/testing/bench): Make progress callback async (#6175)
- fix(std/testing/bench): Clock assertions without --allow-hrtime (#6069)
- refactor(std): Remove testing dependencies from non-test code (#5838)
- refactor(std/http): Rename delCookie to deleteCookie (#6088)
- refactor(std/testing): Rename abbreviated assertions (#6118)
- refactor(std/testing/bench): Remove differentiating on runs count (#6084)

1.0.5 / 2020.06.03

Changes in the CLI:

- fix(fetch): Support 101 status code (#6059)
- fix: REPL BorrowMutError panic (#6055)
- fix: dynamic import BorrowMutError (#6065)
- upgrade: dprint 0.19.1 and swc_ecma_parser 0.24.3 (#6068)
- upgrade: rusty_v8 0.5.0 (#6070)

Changes in std version 0.56.0:

- feat(std/testing): benching progress callback (#5941)
- feat(std/encoding): add base64url module (#5976)
- fix(std/testing/asserts): Format values in assertArrayContains() (#6060)

1.0.4 / 2020.06.02

Changes in the CLI:

- feat(core): Ops can take several zero copy buffers (#4788)
- fix(bundle): better size output (#5997)
- fix(cli): Deno.remove() fails to remove unix socket (#5967)
- fix(cli): compile TS dependencies of JS files (#6000)
- fix(cli): ES private fields parsing in SWC (#5964)
- fix(cli): Better use of @ts-expect-error (#6038)
- fix(cli): media type for .cjs and application/node (#6005)
- fix(doc): remove JSDoc comment truncation (#6031)
- fix(cli/js/web): Body.bodyUsed should use IsReadableStreamDisturbed
- fix(cli/js/web): formData parser for binary files in fetch() (#6015)

- fix(cli/js/web): set null body for null-body status in fetch() (#5980)
- fix(cli/js/web): network error on multiple redirects in fetch() (#5985)
- fix(cli/js/web): Headers.name and FormData.name (#5994)
- upgrade: Rust crates (#5959, #6032)

Changes in std version 0.55.0:

- feat(std/hash): add Sha512 and HmacSha512 (#6009)
- feat(std/http) support code 103 Early Hints (#6021)
- feat(std/http): add TooEarly status code (#5999)
- feat(std/io): add LimitedReader (#6026)
- feat(std/log): buffered file logging (#6014)
- feat(std/mime/multipart): Added multiple FormFile input (#6027)
- feat(std/node): add util.type.isDate (#6029)
- fix(std/http): file server not closing files (#5952)
- fix(std/path): support browsers (#6003)

1.0.3 / 2020.05.29

Changes in the CLI:

- fix: Add unstable checks for Deno.dir and Diagnostics (#5750)
- fix: Add unstable checks for unix transport (#5818)
- fix: Create HTTP cache lazily (#5795)
- fix: Dependency analysis in TS compiler (#5817, #5785, #5870)
- fix: Expose Error.captureStackTrace (#5254)
- fix: Improved typechecking error for unstable props (#5503)
- fix: REPL evaluates in strict mode (#5565)
- fix: Write lock file before running any code (#5794)
- fix(debugger): BorrowMutError when evaluating expression in inspector console (#5822)
- fix(doc): Handle comments at the top of the file (#5891)
- fix(fmt): Handle formatting UTF-8 w/ BOM files (#5881)
- fix(permissions): Fix CWD and exec path leaks (#5642)
- fix(web/blob): DenoBlob name (#5879)
- fix(web/console): Hide `values` for console.table if display not necessary (#5914)
- fix(web/console): Improve indentation when displaying objects with console.log (#5909)
- fix(web/encoding): atob should throw dom exception (#5730)
- fix(web/fetch): Make Response constructor standard (#5787)
- fix(web/fetch): Allow ArrayBuffer as Fetch request body (#5831)
- fix(web/formData): Set default filename for Blob to (#5907)
- upgrade: dprint to 0.19.0 (#5899)

Changes in std version 0.54.0:

- feat(std/encoding): Add base64 (#5811)
- feat(std/http): Handle .wasm files in file_server (#5896)
- feat(std/node): Add link/linkSync polyfill (#5930)
- feat(std/node): fs.writeFile/sync path can now be an URL (#5652)
- feat(std/testing): Return results in benchmark promise (#5842)
- fix(std/http): readTrailer evaluates header names by case-insensitive (#4902)
- fix(std/log): Improve the calculation of byte length (#5819)
- fix(std/log): Fix FileHandler test with mode 'x' on non-English systems (#5757)
- fix(std/log): Use writeAllSync instead of writeSync (#5868)

- fix(std/testing/asserts): Support browsers (#5847)

1.0.2 / 2020.05.22

Changes in the CLI:

- fix: --inspect flag working like --inspect-brk (#5697)
- fix: Disallow http imports for modules loaded over https (#5680)
- fix: Redirects handling in module analysis (#5726)
- fix: SWC lexer settings and silent errors (#5752)
- fix: TS type imports (#5733)
- fix(fmt): Do not panic on new expr with no parens. (#5734)
- fix(cli/js/streams): High water mark validation (#5681)

Changes in std version 0.53.0:

- fix(std/http): file_server's target directory (#5695)
- feat(std/hash): add md5 (#5719)
- refactor: Move std/fmt/sprintf.ts to std/fmt/printf.ts (#4567)

1.0.1 / 2020.05.20

Changes in the CLI:

- fix(doc): crash on formatting type predicate (#5651)
- fix: Implement Deno.kill for windows (#5347)
- fix: Implement Deno.symlink() for windows (#5533)
- fix: Make Deno.remove() work with directory symlinks on windows (#5488)
- fix: Mark Deno.pid and Deno.noColor as const (#5593)
- fix: Remove debug prints introduced in e18aaf49c (#5356)
- fix: Return error if more than one listener calls `WorkerHandle::get_event()` (#5461)
- fix: Simplify fmt::Display for ModuleResolutionError (#5550)
- fix: URL utf8 encoding (#5557)
- fix: don't panic on Deno.close invalid argument (#5320)
- fix: panic if DENO_DIR is a relative path (#5375)
- fix: setTimeout and friends have too strict types (#5412)
- refactor: rewrite TS dependency analysis in Rust (#5029, #5603)
- update: dprint 0.18.4 (#5671)

Changes in std version 0.52.0:

- feat(std/bytes): add hasSuffix and contains functions, update docs (#4801)
- feat(std/fmt): rgb24 and bgRgb24 can use numbers for color (#5198)
- feat(std/hash): add fnv implementation (#5403)
- feat(std/node) Export TextDecoder and TextEncoder from util (#5663)
- feat(std/node): Add fs.promises.readFile (#5656)
- feat(std/node): add util.callbackify (#5415)
- feat(std/node): first pass at url module (#4700)
- feat(std/node): fs.writeFileSync polyfill (#5414)
- fix(std/hash): SHA1 hash of Uint8Array (#5086)
- fix(std/http): Add .css to the MEDIA_TYPES. (#5367)
- fix(std/io): BufReader should not share the internal buffer across reads (#4543)
- fix(std/log): await default logger setup (#5341)
- fix(std/node) improve fs.close compatibility (#5649)
- fix(std/node): fs.readFile should take string as option (#5316)

- fix(std/testing): Provide message and diff for assertStrictEq (#5417)

1.0.0 / 2020.05.13

Read more about this release at https://deno.land/v1

- fix: default to 0.0.0.0 for Deno.listen (#5203)
- fix: Make --inspect-brk pause on the first line of *user* code (#5250)
- fix: Source maps in inspector for local files (#5245)
- upgrade: TypeScript 3.9 (#4510)

1.0.0-rc3 / 2020.05.12

- BREAKING: Remove public Rust API for the "deno" crate (#5226)
- feat(core): Allow starting isolate from snapshot bytes on the heap (#5187)
- fix: Check permissions in SourceFileFetcher (#5011)
- fix: Expose ErrorEvent globally (#5222)
- fix: Remove default --allow-read perm for deno test (#5208)
- fix: Source maps in inspector (#5223)
- fix(std/encoding/yaml): Correct exports (#5191)
- fix(plugins): prevent segfaults on windows (#5210)
- upgrade: dprint 0.17.2 (#5195)

1.0.0-rc2 / 2020.05.09

- BREAKING(std): Reorg modules, mark as unstable (#5087, #5177)
- BREAKING(std): Revert "Make WebSocket Reader/Writer" (#5002, #5141)
- BREAKING: Deno.execPath should require allow-read (#5109)
- BREAKING: Make Deno.hostname unstable #5108
- BREAKING: Make Worker with Deno namespace unstable (#5128)
- BREAKING: Remove support for .wasm imports (#5135)
- feat(bundle): Add --config flag (#5130)
- feat(bundle): Format output (#5139)
- feat(doc): Handle default exports (#4873)
- feat(repl): Add hint on how to exit REPL (#5143)
- feat(std/fmt): add 8bit and 24bit ANSI colors (#5168)
- feat(std/node): add fs.writefile / fs.promises.writeFile (#5054)
- feat(upgrade): Allow specifying a version (#5156)
- feat(workers): "crypto" global accessible in Worker scope (#5121)
- feat: Add support for X-Deno-Warning header (#5161)
- fix(imports): Fix panic on unsupported scheme (#5131)
- fix(inspector): Fix inspector hanging when task budget is exceeded (#5083)
- fix: Allow multiple Set-Cookie headers (#5100)
- fix: Better error message when DENO_DIR can't be created (#5120)
- fix: Check destination length in encodeInto in TextEncoder (#5078)
- fix: Correct type error text (#5150)
- fix: Remove unnecessary ProcessStdio declaration (#5092)
- fix: unify display of errors from Rust and JS (#5183)
- upgrade: rust crates (#5104)
- upgrade: to rusty_v8 0.4.2 / V8 8.4.300 (#5113)

v1.0.0-rc1 / 2020.05.04

- BREAKING: make WebSocket directly implement AsyncIterable (#5045)
- BREAKING: remove CLI 'deno script.ts' alias to 'deno run script.ts' (#5026)

- BREAKING: remove support for JSON imports (#5037)
- BREAKING: remove window.location and self.location (#5034)
- BREAKING: reorder std/io/utils copyBytes arguments (#5022, #5021)
- feat(URL): Support drive letters for file URLs on Windows (#5074)
- feat(deno install): simplify CLI flags (#5036)
- feat(deno fmt): Add `deno-fmt-ignore` and `deno-fmt-ignore-file` comment support #5075
- feat(std): Add sha256 and sha224 support (along with HMAC variants) (#5066)
- feat(std/node): ability add to path argument to be URL type (#5055)
- feat(std/node): make process global (#4985)
- feat(std/node): toString for globals (#5013)
- feat: Add WritableStreams, TransformStream, TransformStreamController (#5042, #4980)
- feat: Make WebSocket Reader/Writer (#5002)
- feat: make Deno.cwd stable (#5068)
- fix(console): Formatting misalignment on console.table (#5046)
- fix(deno doc): Better repr for object literal types (#4998)
- fix(deno fmt): Format `abstract async` as `abstract async` (#5020)
- fix(std): Use fromFileUrl (#5005)
- fix(std/http): Hang when content-length unhandled (#5024)
- fix: Deno.chdir Should require allow-read not allow-write (#5033)
- fix: Respect NO_COLOR for stack frames (#5051)
- fix: URL constructor throws confusing error on invalid scheme (#5057)
- fix: Disallow static import of local modules from remote modules (#5050)
- fix: Misaligned error reporting on tab char (#5032)
- refactor(core): Add "prepare_load" hook to ModuleLoader trait (#4866)
- refactor: Don't expose unstable APIs to runtime (#5061 #4957)

v0.42.0 / 2020.04.29

- BREAKING: "address" renamed to "path" in UnixAddr/UnixConnectOptions/UnixListenOptions (#4959)
- BREAKING: Change DirEntry to not require extra stat syscall (#4941)
- BREAKING: Change order of args in Deno.copy() (#4885)
- BREAKING: Change order of copyN arguments (#4900)
- BREAKING: Change return type of Deno.resources() (#4893)
- BREAKING: Deno.chdir() should require --allow-write (#4889)
- BREAKING: Factor out Deno.listenDatagram(), mark as unstable (#4968)
- BREAKING: Make shutdown unstable and async (#4940)
- BREAKING: Make unix sockets require allow-write (#4939)
- BREAKING: Map-like interface for Deno.env (#4942)
- BREAKING: Mark --importmap as unstable (#4934)
- BREAKING: Mark Deno.dir() unstable (#4924)
- BREAKING: Mark Deno.kill() as unstable (#4950)
- BREAKING: Mark Deno.loadavg() and osRelease() as unstable (#4938)
- BREAKING: Mark Deno.setRaw() as unstable (#4925)
- BREAKING: Mark Deno.umask() unstable (#4935)
- BREAKING: Mark Deno.utime() as unstable (#4955)
- BREAKING: Mark runtime compile ops as unstable (#4912)
- BREAKING: Mark signal APIs as unstable (#4926)
- BREAKING: Remove Conn.closeRead (#4970)
- BREAKING: Remove Deno.EOF, use null instead (#4953)
- BREAKING: Remove Deno.OpenMode (#4884)
- BREAKING: Remove Deno.runTests() API (#4922)

- BREAKING: Remove Deno.symbols namespace (#4936)
- BREAKING: Remove combined io interface like ReadCloser (#4944)
- BREAKING: Remove overload of Deno.test() taking named function (#4951)
- BREAKING: Rename Deno.fsEvents() to Deno.watchFs() (#4886)
- BREAKING: Rename Deno.toAsyncIterator() to Deno.iter() (#4848)
- BREAKING: Rename FileInfo time fields and represent them as Date objects (#4932)
- BREAKING: Rename SeekMode variants to camelCase and stabilize (#4946)
- BREAKING: Rename TLS APIs to camel case (#4888)
- BREAKING: Use LLVM target triple for Deno.build (#4948)
- BREAKING: introduce unstable flag; mark Deno.openPlugin, link, linkSync, symlink, symlinkSync as unstable (#4892)
- BREAKING: make camel case readDir, readLink, realPath (#4995)
- BREAKING: remove custom implementation of Deno.Buffer.toString() (#4992)
- BREAKING: std/node: require_ -> require (#4828)
- feat(fmt): parallelize formatting (#4823)
- feat(installer): Add DENO_INSTALL_ROOT (#4787)
- feat(std/http): Improve parseHTTPVersion (#4930)
- feat(std/io): Increase copyN buffer size to match go implementation (#4904)
- feat(std/io): synchronous buffered writer (#4693)
- feat(std/path): Add fromFileUrl() (#4993)
- feat(std/uuid): Implement uuid v5 (#4916)
- feat(test): add quiet flag (#4894)
- feat: Add Deno.readTextFile(), Deno.writeTextFile(), with sync counterparts (#4901)
- feat: Add buffer size argument to copy (#4907)
- feat: Add close method to Plugin (#4670) (#4785)
- feat: Change URL.port implementation to match WHATWG specifications (#4954)
- feat: Deno.startTLS() (#4773, #4965)
- feat: Make zero a valid port for URL (#4963)
- feat: add help messages to Deno.test() sanitizers (#4887)
- feat: support Deno namespace in Worker API (#4784)
- fix(core): Op definitions (#4814)
- fix(core): fix top-level-await error handling (#4911)
- fix(core/js_errors): Get error's name and message from JS fields (#4808)
- fix(format): stdin not formatting JSX (#4971)
- fix(installer): handle case-insensitive uri (#4909)
- fix(std): existsFile test
- fix(std/fs): move dest if not exists and overwrite (#4910)
- fix(std/io): Make std/io copyN write the whole read buffer (#4978)
- fix(std/mime): MultipartReader for big files (#4865)
- fix(std/node): bug fix and tests fs/mkdir (#4917)
- fix: bug in Deno.copy (#4977)
- fix: don't throw RangeError when an invalid date is passed (#4929)
- fix: make URLSearchParams more standardized (#4695)
- refactor(cli): Improve source line formatting (#4832)
- refactor(cli): Move resource_table from deno::State to deno_core::Isolate (#4834)
- refactor(cli): Remove bootstrap methods from global scope after bootstrapping (#4869)
- refactor(cli/doc): Factor out AstParser from DocParser (#4923)
- refactor(cli/inspector): Store debugger url on DenoInspector (#4793)
- refactor(cli/js): Rewrite streams (#4842)
- refactor(cli/js/io): Change type of stdio handles in JS api (#4891, #4952)

- refactor(cli/js/io): Rename sync io interfaces (#4945)
- refactor(cli/js/net): Deno.listener closes when breaking out of async iterator (#4976)
- refactor(cli/js/permissions): Split read and write permission descriptors (#4774)
- refactor(cli/js/testing): Rename disableOpSanitizer to sanitizeOps (#4854)
- refactor(cli/js/web): Change InspectOptions, mark Deno.inspect as stable (#4967)
- refactor(cli/js/web): Decouple Console implementation from stdout (#4899)
- refactor(cli/ops): Replace block_on in net interfaces (#4796)
- refactor(cli|std): Add no-async-promise-executor lint rule (#4809)
- refactor(core): Modify op dispatcher to include &mut Isolate argument (#4821)
- refactor(core): Remove core/plugin.rs (#4824)
- refactor(core): Rename deno_core::Isolate to deno_core::CoreIsolate (#4851)
- refactor(core): add id field to RecursiveModuleLoad (#4905)
- refactor(std/log): support enum log level (#4859)
- refactor(std/node): proper Node polyfill directory iteration (#4783)
- upgrade: Rust 1.43.0 (#4871)
- upgrade: dprint 0.13.0 (#4816)
- upgrade: dprint 0.13.1 (#4853)
- upgrade: rusty_v8 v0.4.0 (#4856)
- chore: Mark Deno.Metrics and Deno.RunOptions as stable (#4949)

v0.41.0 / 2020.04.16

- BREAKING: Improve readdir() and FileInfo interfaces (#4763)
- BREAKING: Remove deprecated APIs for mkdir and mkdirSync (#4615)
- BREAKING: Make fetch API more web compatible (#4687)
- BREAKING: Remove std/testing/format.ts (#4749)
- BREAKING: Migrate std/types to deno.land/x/types/ (#4713, #4771)
- feat(doc): support for runtime built-ins (#4635)
- feat(std/log): rotating handler, docs improvements (#4674)
- feat(std/node): Add isPrimitive method (#4673)
- feat(std/node/fs): Add copyFile and copyFileSync methods (#4726)
- feat(std/signal): Add onSignal method (#4696)
- feat(std/testing): Change output of diff (#4697)
- feat(std/http): Verify cookie name (#4685)
- feat(std/multipart): Make readForm() type safe (#4710)
- feat(std/uuid): Add UUID v1 (#4758)
- feat(install): Honor log level arg (#4714)
- feat(workers): Make Worker API more web compatible (#4684, #4734, #4391, #4737, #4746)
- feat: Add AbortController and AbortSignal API (#4757)
- fix(install): Clean up output on Windows (#4764)
- fix(core): Handle SyntaxError during script compilation (#4770)
- fix(cli): Async stack traces and stack formatting (#4690, #4706, #4715)
- fix(cli): Remove unnecessary namespaces in "deno types" (#4683, #4698, #4718, #4719, #4736, #4741)
- fix(cli): Panic on invalid UTF-8 string (#4704)
- fix(cli/js/net): Make generator return types iterable (#4661)
- fix(doc): Handle optional and extends fields (#4738, #4739)
- refactor: Event and EventTarget implementation (#4707)
- refactor: Use synchronous syscalls where applicable (#4762)
- refactor: Remove calls to futures::executor::block_on (#4760, #4775)
- upgrade: Rust crates (#4742)

v0.40.0 / 2020.04.08

- BREAKING: Rename 'deno fetch' subcommand to 'deno cache' (#4656)
- BREAKING: Remove std/testing/runner.ts (#4649)
- feat(std/flags): Pass key and value to unknown (#4637)
- feat(std/http): Respond with 400 on request parse failure (#4614)
- feat(std/node): Add exists and existsSync (#4655)
- feat: Add File support in FormData (#4632)
- feat: Expose ReadableStream and make Blob more standardized (#4581)
- feat: add --importmap flag to deno bundle (#4651)
- fix(#4546): Added Math.trunc to toSecondsFromEpoch to conform the result to u64 (#4575)
- fix(file_server): use text/typescript instead of application/typescript (#4620)
- fix(std/testing): format bigint (#4626)
- fix: Drop headers with trailing whitespace in header name (#4642)
- fix: Fetch reference types for JS files (#4652)
- fix: Improve deno doc (#4672, #4625)
- fix: On init create disk_cache directory if it doesn't already exists (#4617)
- fix: Remove unnecessary namespaces in "deno types" (#4677, #4675, #4669, #4668, #4665, #4663, #4662)
- upgrade: Rust crates (#4679)

v0.39.0 / 2020.04.03

- BREAKING CHANGE: Move encode, decode helpers to /std/encoding/utf8.ts, delete /std/strings/ (#4565)
- BREAKING CHANGE: Remove /std/media_types (#4594)
- BREAKING CHANGE: Remove old release files (#4545)
- BREAKING CHANGE: Remove std/strings/pad.ts because String.prototype.padStart exists (#4564)
- feat: Add common to std/path (#4527)
- feat: Added colors to doc output (#4518)
- feat: Expose global state publicly (#4572)
- feat: Make inspector more robust, add --inspect-brk support (#4552)
- feat: Publish deno types on release (#4583)
- feat: Support dynamic import in bundles. (#4561)
- feat: deno test --filter (#4570)
- feat: improve console.log serialization (#4524, #4472)
- fix(#4550): setCookie should append cookies (#4558)
- fix(#4554): use --inspect in repl & eval (#4562)
- fix(deno doc): handle 'declare' (#4573)
- fix(deno doc): parse super-class names (#4595)
- fix(deno doc): parse the "implements" clause of a class def (#4604)
- fix(file_server): serve appropriate content-type header (#4555)
- fix(inspector): proper error message on port collision (#4514)
- fix: Add check to fail the benchmark test on server error (#4519)
- fix: Properly handle invalid utf8 in paths (#4609)
- fix: async ops sanitizer false positives in timers (#4602)
- fix: invalid blob type (#4536)
- fix: make Worker.poll private (#4603)
- fix: remove `Send` trait requirement from the `Resource` trait (#4585)
- refactor(testing): Reduce testing interfaces (#4451)
- upgrade: dprint to 0.9.10 (#4601)
- upgrade: rusty_v8 v0.3.10 (#4576)

v0.38.0 / 2020.03.28

- feat: Add "deno doc" subcommand (#4500)

- feat: Support --inspect, Chrome Devtools support (#4484)
- feat: Support Unix Domain Sockets (#4176)
- feat: add queueMicrotask to d.ts (#4477)
- feat: window.close() (#4474)
- fix(console): replace object abbreviation with line breaking (#4425)
- fix: add fsEvent notify::Error casts (#4488)
- fix: hide source line if error message longer than 150 chars (#4487)
- fix: parsing bug (#4483)
- fix: remove extra dot in Permission request output (#4471)
- refactor: rename ConsoleOptions to InspectOptions (#4493)
- upgrade: dprint 0.9.6 (#4509, #4491)
- upgrade: prettier 2 for internal code formatting (#4498)
- upgrade: rusty_v8 to v0.3.9 (#4505)

v0.37.1 / 2020.03.23

- fix: Statically link the C runtime library on Windows (#4469)

v0.37.0 / 2020.03.23

- BREAKING CHANGE: FileInfo.len renamed to FileName.size (#4338)
- BREAKING CHANGE: Rename Deno.run's args to cmd (#4444)
- feat(ci): Releases should all use zip and LLVM target triples (#4460)
- feat(console): Symbol.toStringTag and display Object symbol entries (#4388)
- feat(std/node): Add chmod Node polyfill (#4358)
- feat(std/node): Add node querystring polyfill (#4370)
- feat(std/node): Node polyfill for fs.chown and fs.close (#4377)
- feat(std/permissions): Add helper functions for permissions to std (#4258)
- feat(std/types): Provide types for React and ReactDOM (#4376)
- feat(test): Add option to skip tests (#4351)
- feat(test): Add support for jsx/tsx for deno test (#4369)
- feat: Add mode option to open/create (#4289)
- feat: Deno.test() sanitizes ops and resources (#4399)
- feat: Fetch should accept a FormData body (#4363)
- feat: First pass at "deno upgrade" (#4328)
- feat: Prvode way to build Deno without building V8 from source (#4412)
- feat: Remove `Object.prototype.__proto__` (#4341)
- fix(std/http): Close open connections on server close (#3679)
- fix(std/http): Properly await ops in a server test (#4436)
- fix(std/http): Remove bad error handling (#4435)
- fix(std/node): Node polyfill fsAppend rework (#4322)
- fix(std/node): Stack traces for modules imported via require (#4035)
- fix: Importing JSON doesn't work in bundles (#4404)
- fix: Simplify timer with macrotask callback (#4385)
- fix: Test runner ConnectionReset bug (#4424)
- fix: chmod should throw on Windows (#4446)
- fix: fetch closes unused body (#4393)
- perf: Optimize TextEncoder and TextDecoder (#4430, #4349)
- refactor: Improve test runner (#4336, #4352, #4356, #4371)
- refactor: Remove std/testing/runner.ts, use deno test (#4397, #4392)
- upgrade: Rust 1.42.0 (#4331)
- upgrade: Rust crates (#4412)

- upgrade: to rusty_v8 0.3.5 / v8 8.2.308 (#4364)

v0.36.0 / 2020.03.11
- BREAKING CHANGE: Remove Deno.errors.Other (#4249)
- BREAKING CHANGE: Rename readDir -> readdir (#4225)
- feat(std/encoding): add binary module (#4274)
- feat(std/node): add appendFile and appendFileSync (#4294)
- feat(std/node): add directory classes (#4087)
- feat(std/node): add os.tmpdir() implementation (#4213)
- feat: Add Deno.umask (#4290)
- feat: Add global --quiet flag (#4135)
- feat: Improvements to std/flags. (#4279)
- feat: Make internal error frames dimmer (#4201)
- feat: Support async function and EventListenerObject as listeners (#4240)
- feat: add actual error class to fail message (#4305)
- feat: seek should return cursor position (#4211)
- feat: support permission mode in mkdir (#4286)
- feat: update metrics to track different op types (#4221)
- fix: Add content type for wasm, fix encoding in wasm test fixture (#4269)
- fix: Add waker to StreamResource to fix hang on close bugs (#4293)
- fix: Flattens dispatch error handling to produce one less useless stack frame on op errors. (#4189)
- fix: JavaScript dependencies in bundles. (#4215)
- fix: Stricter permissions for Deno.makeTemp (#4318)
- fix: `deno install` file name including extra dot on Windows (#4243)
- fix: inlining of lib.dom.iterable.d.ts. (#4242)
- fix: properly close FsEventsResource (#4266)
- fix: remove unwanted ANSI Reset Sequence (#4268)
- perf: use Object instead of Map for promise table (#4309)
- perf: use subarray instead of slice in dispatch minimal (#4180)
- refactor(cli/js): add assertOps and assertResources sanitizer in cli/js/ unit tests (#4209, #4161)
- refactor(cli/js/net): Cleanup iterable APIs (#4236)
- refactor(core): improve exception handling(#4214, #4214, #4198)
- refactor(core): rename structures related to Modules (#4217)
- refactor: Cleanup options object parameters (#4296)
- refactor: Migrate internal bundles to System (#4233)
- refactor: Rename Option -> Options (#4226)
- refactor: cleanup compiler runtimes (#4230)
- refactor: preliminary cleanup of Deno.runTests() (#4237)
- refactor: reduce unnecessary output in cli/js tests (#4182)
- refactor: reorganize cli/js (#4317, #4316, #4310, #4250, #4302, #4283, #4264)
- refactor: rewrite testPerm into unitTest (#4231)
- refactor: uncomment tests broken tests, use skip (#4311)
- upgrade: dprint 0.8.0 (#4308, #4314)
- upgrade: rust dependencies (#4270)
- upgrade: typescript 3.8.3 (#4301)

v0.35.0 / 2020.02.28
- feat: Deno.fsEvents() (#3452)
- feat: Support UDP sockets (#3946)
- feat: Deno.setRaw(rid, mode) to turn on/off raw mode (#3958)

- feat: Add Deno.formatDiagnostics (#4032)
- feat: Support TypeScript eval through `deno eval -T` flag (#4141)
- feat: Support types compiler option in compiler APIs (#4155)
- feat: add std/examples/chat (#4022, #4109, #4091)
- feat: support brotli compression for fetch API (#4082)
- feat: reverse URL lookup for cache (#4175)
- feat(std/node): add improve os module (#4064, #4075, #4065)
- feat(std/node): add os Symbol.toPrimitive methods (#4073)
- fix(fetch): proper error for unsupported protocol (#4085)
- fix(std/examples): add tests for examples (#4094)
- fix(std/http): Consume unread body before reading next request (#3990)
- fix(std/ws): createSecKey logic (#4063)
- fix(std/ws): provide default close code for ws.close() (#4172)
- fix(std/ws): sock shouldn't throw eof error when failed to read frame (#4083)
- fix: Bundles can be sync or async based on top level await (#4124)
- fix: Move WebAsssembly namespace to shared_globals (#4084)
- fix: Resolve makeTemp paths from CWD (#4104)
- fix: Return non-zero exit code on malformed stdin fmt (#4163)
- fix: add window.self read-only property (#4131)
- fix: fetch in workers (#4054)
- fix: fetch_cached_remote_source support redirect URL without base (#4099)
- fix: issues with JavaScript importing JavaScript. (#4120)
- fix: rewrite normalize_path (#4143)
- refactor(std/http): move io functions to http/io.ts (#4126)
- refactor: Deno.errors (#3936, #4058, #4113, #4093)
- upgrade: TypeScript 3.8 (#4100)
- upgrade: dprint 0.7.0 (#4130)
- upgrade: rusty_v8 0.3.4 (#4179)

v0.34.0 / 2020.02.20

- feat: Asynchronous event iteration node polyfill (#4016)
- feat: Deno.makeTempFile (#4024)
- feat: Support loading additional TS lib files (#3863)
- feat: add --cert flag for http client (#3972)
- feat(std/io): Export readDelim(), readStringDelim() and readLines() from bufio.ts (#4019)
- fix(deno test): support directories as arguments (#4011)
- fix: Enable TS strict mode by default (#3899)
- fix: detecting AMD like imports (#4009)
- fix: emit when bundle contains single module (#4042)
- fix: mis-detecting imports on JavaScript when there is no checkJs (#4040)
- fix: skip non-UTF-8 dir entries in Deno.readDir() (#4004)
- refactor: remove run_worker_loop (#4028)
- refactor: rewrite file_fetcher (#4037, #4030)
- upgrade: dprint 0.6.0 (#4026)

v0.33.0 / 2020.02.13

- feat(std/http): support trailer headers (#3938, #3989)
- feat(std/node): Add readlink, readlinkSync (#3926)
- feat(std/node): Event emitter node polyfill (#3944, #3959, #3960)
- feat(deno install): add --force flag and remove yes/no prompt (#3917)

- feat: Improve support for diagnostics from runtime compiler APIs (#3911)
- feat: `deno fmt -` formats stdin and print to stdout (#3920)
- feat: add std/signal (#3913)
- feat: make testing API built-in Deno.test() (#3865, #3930, #3973)
- fix(std/http): align serve and serveTLS APIs (#3881)
- fix(std/http/file_server): don't crash on "%" pathname (#3953)
- fix(std/path): Use non-capturing groups in globrex() (#3898)
- fix(deno types): don't panic when piped to head (#3910)
- fix(deno fmt): support top-level await (#3952)
- fix: Correctly determine a --cached-only error (#3979)
- fix: No longer require aligned buffer for shared queue (#3935)
- fix: Prevent providing --allow-env flag twice (#3906)
- fix: Remove unnecessary EOF check in Deno.toAsyncIterable (#3914)
- fix: WASM imports loaded HTTP (#3856)
- fix: better WebWorker API compatibility (#3828)
- fix: deno fmt improvements (#3988)
- fix: make WebSocket.send() exclusive (#3885)
- refactor: Improve `deno bundle` by using System instead of AMD (#3965)
- refactor: Remove conditionals from installer (#3909)
- refactor: peg workers to a single thread (#3844, #3968, #3931, #3903, #3912, #3907, #3904)

v0.32.0 / 2020.02.03

- BREAKING CHANGE: Replace formatter for "deno fmt", use dprint (#3820, #3824, #3842)
- BREAKING CHANGE: Remove std/prettier (#3820)
- BREAKING CHANGE: Remove std/installer (#3843)
- BREAKING CHANGE: Remove --current-thread flag (#3830)
- BREAKING CHANGE: Deno.makeTempDir() checks permissions (#3810)
- feat: deno install in Rust (#3806)
- feat: Improve support of type definitions (#3755)
- feat: deno fetch supports --lock-write (#3787)
- feat: deno eval supports --v8-flags=… (#3797)
- feat: descriptive permission errors (#3808)
- feat: Make fetch API more standards compliant (#3667)
- feat: deno fetch supports multiple files (#3845)
- feat(std/node): Endianness (#3833)
- feat(std/node): Partial os polyfill (#3821)
- feat(std/examples): Bring back xeval (#3822)
- feat(std/encoding): Add base32 support (#3855)
- feat(deno_typescript): Support crate imports (#3814)
- fix: Panic on cache miss (#3784)
- fix: Deno.remove() to properly remove dangling symlinks (#3860)
- refactor: Use tokio::main attribute in lib.rs (#3831)
- refactor: Provide TS libraries for window and worker scope (#3771, #3812, #3728)
- refactor(deno_core): Error tracking and scope passing (#3783)
- refactor(deno_core): Rename PinnedBuf to ZeroCopyBuf (#3782)
- refactor(deno_core): Change Loader trait (#3791)
- upgrade: Rust 1.41.0 (#3838)
- upgrade: Rust crates (#3829)

v0.31.0 / 2020.01.24

- BREAKING CHANGE: remove support for blob: URL in Worker (#3722)
- BREAKING CHANGE: remove Deno namespace support and noDenoNamespace option in Worker constructor (#3722)
- BREAKING CHANGE: rename dial to connect and dialTLS to connectTLS (#3710)
- feat: Add signal handlers (#3757)
- feat: Implemented alternative open mode in files (#3119)
- feat: Use globalThis to reference global scope (#3719)
- feat: add AsyncUnref ops (#3721)
- feat: stabilize net Addr (#3709)
- fix: correct yaml's sortKeys type (#3708)
- refactor: Improve path handling in permission checks (#3714)
- refactor: Improve web workers (#3722, #3732, #3730, #3735)
- refactor: Reduce number of ErrorKind variants (#3662)
- refactor: Remove Isolate.shared_response_buf optimization (#3759)
- upgrade: rusty_v8 (#3764, #3769, #3741)

v0.30.0 / 2020.01.17

- BREAKING CHANGE Revert "feat(flags): script arguments come after '--'" (#3681)
- feat(fs): add more unix-only fields to FileInfo (#3680)
- feat(http): allow response body to be string (#3705)
- feat(std/node): Added node timers builtin (#3634)
- feat: Add Deno.symbols and move internal fields for test (#3693)
- feat: Add gzip, brotli and ETag support for file fetcher (#3597)
- feat: support individual async handler for each op (#3690)
- fix(workers): minimal error handling and async module loading (#3665)
- fix: Remove std/multipart (#3647)
- fix: Resolve read/write whitelists from CWD (#3684)
- fix: process hangs when fetch called (#3657)
- perf: Create an old program to be used in snapshot (#3644, #3661)
- perf: share http client in file fetcher (#3683)
- refactor: remove Isolate.current_send_cb_info and DenoBuf, port Isolate.shared_response_buf (#3643)

v0.29.0 / 2020.01.09

- BREAKING CHANGE Remove xeval subcommand (#3630)
- BREAKING CHANGE script arguments should come after '--' (#3621)
- BREAKING CHANGE Deno.mkdir should conform to style guide BREAKING CHANGE (#3617)
- BREAKING CHANGE Deno.args only includes script args (#3628)
- BREAKING CHANGE Rename crates: 'deno' to 'deno_core' and 'deno_cli' to 'deno' (#3600)
- feat: Add Deno.create (#3629)
- feat: Add compiler API (#3442)
- fix(ws): Handshake with correctly empty search string (#3587)
- fix(yaml): Export parseAll (#3592)
- perf: TextEncoder.encode improvement (#3596, #3589)
- refactor: Replace libdeno with rusty_v8 (#3556, #3601, #3602, #3605, #3611, #3613, #3615)
- upgrade: V8 8.1.108 (#3623)

v0.28.1 / 2020.01.03

- feat(http): make req.body a Reader (#3575)
- fix: dynamically linking to OpenSSL (#3586)

v0.28.0 / 2020.01.02

- feat: Add Deno.dir("executable") (#3526)
- feat: Add missing mod.ts files in std (#3509)
- fix(repl): Do not crash on async op reject (#3527)
- fix(std/encoding/yaml): support document separator in parseAll (#3535)
- fix: Allow reading into a 0-length array (#3329)
- fix: Drop unnecessary Object.assign from createResolvable() (#3548)
- fix: Expose shutdown() and ShutdownMode TS def (#3558, #3560)
- fix: Remove wildcard export in uuid module (#3540)
- fix: Return null on error in Deno.dir() (#3531)
- fix: Use shared HTTP client (#3563)
- fix: Use sync ops when clearing the console (#3533)
- refactor: Move HttpBody to cli/http_util.rs (#3569)
- upgrade: Reqwest to 0.10.0 (#3567)
- upgrade: Rust to 1.40.0 (#3542)
- upgrade: Tokio 0.2 (#3418, #3571)

v0.27.0 / 2019.12.18

- feat: Support utf8 in file_server (#3495)
- feat: add help & switch to flags to file_server (#3489)
- feat: fetch should support URL instance as input (#3496)
- feat: replace Deno.homeDir with Deno.dir (#3491, #3518)
- feat: show detailed version with --version (#3507)
- fix(installer): installs to the wrong directory on Windows (#3462)
- fix(std/http): close connection on .respond() error (#3475)
- fix(std/node): better error message for read perm in require() (#3502)
- fix(timer): due/now Math.max instead of min (#3477)
- fix: Improve empty test case error messages (#3514)
- fix: Only swallow NotFound errors in std/fs/expandGlob() (#3479)
- fix: decoding uri in file_server (#3187)
- fix: file_server should get file and fileInfo concurrently (#3486)
- fix: file_server swallowing permission errors (#3467)
- fix: isolate tests silently failing (#3459)
- fix: permission errors are swallowed in fs.exists, fs.emptyDir, fs.copy (#3493, #3501, #3504)
- fix: plugin ops should change op count metrics (#3455)
- fix: release assets not being executable (#3480)
- upgrade: tokio 0.2 in deno_core_http_bench, take2 (#3435)
- upgrade: upgrade subcommand links to v0.26.0 (#3492)

v0.26.0 / 2019.12.05

- feat: Add --no-remote, rename --no-fetch to --cached-only (#3417)
- feat: Native plugins AKA dlopen (#3372)
- fix: Improve html for file_server (#3423)
- fix: MacOS Catalina build failures (#3441)
- fix: Realpath behavior in windows (#3425)
- fix: Timer/microtask ordering (#3439)
- fix: Tweaks to arg_hacks and add v8-flags to repl (#3409)
- refactor: Disable eager polling for ops (#3434)

v0.25.0 / 2019.11.26

- feat: Support named exports on bundles (#3352)
- feat: Add --check for deno fmt (#3369)
- feat: Add Deno.realpath (#3404)
- feat: Add ignore parser for std/prettier (#3399)
- feat: Add std/encoding/yaml module (#3361)
- feat: Add std/node polyfill for require() (#3382, #3380)
- feat: Add std/node/process (#3368)
- feat: Allow op registration during calls in core (#3375)
- feat: Better error message for missing module (#3402)
- feat: Support load yaml/yml prettier config (#3370)
- fix: Make private namespaces in lib.deno_runtime.d.ts more private (#3400)
- fix: Remote .wasm import content type issue (#3351)
- fix: Run std tests with cargo test (#3344)
- fix: deno fmt should respect prettierrc and prettierignore (#3346)
- fix: std/datetime toIMF bug (#3357)
- fix: better error for 'relative import path not prefixed with...' (#3405)
- refactor: Elevate DenoPermissions lock to top level (#3398)
- refactor: Reorganize flags, removes ability to specify run arguments like `--allow-net` after the script (#3389)
- refactor: Use futures 0.3 API (#3358, #3359, #3363, #3388, #3381)
- chore: Remove unneeded tokio deps (#3376)

v0.24.0 / 2019.11.14

- feat: Add Node compat module std/node (#3319)
- feat: Add permissions.request (#3296)
- feat: Add prettier flags to deno fmt (#3314)
- feat: Allow http server to take { hostname, port } argument (#3233)
- feat: Make bundles fully standalone (#3325)
- feat: Support .wasm via imports (#3328)
- fix: Check for closing status when iterating Listener (#3309)
- fix: Error handling in std/fs/walk() (#3318)
- fix: Exclude prebuilt from deno_src release (#3272)
- fix: Turn on TS strict mode for deno_typescript (#3330)
- fix: URL parse bug (#3316)
- refactor: resources and workers (#3285, #3271, #3274, #3342, #3290)
- upgrade: Prettier 1.19 (#3275, #3305)
- upgrade: Rust deps (#3292)
- upgrade: TypeScript 3.7 (#3275)
- upgrade: V8 8.0.192

v0.23.0 / 2019.11.04

- feat: Add serveTLS and listenAndServeTLS (#3257)
- feat: Lockfile support (#3231)
- feat: Adds custom inspect method for URL (#3241)
- fix: Support for deep `Map` equality with `asserts#equal` (#3236, #3258)
- fix: Make EOF unique symbol (#3244)
- fix: Prevent customInspect error from crashing console (#3226)

v0.22.0 / 2019.10.28

- feat: Deno.listenTLS (#3152)

- feat: Publish source tarballs for releases (#3203)
- feat: Support named imports/exports for subset of properties in JSON modules (#3210)
- feat: Use web standard Permissions API (#3200)
- feat: Remove --no-prompt flag, fail on missing permissions (#3183)
- feat: top-level-for-await (#3212)
- feat: Add ResourceTable in core (#3150)
- feat: Re-enable standard stream support for fetch bodies (#3192)
- feat: Add CustomInspect for Headers (#3130)
- fix: Cherry-pick depot_tools 6a1d778 to fix macOS Catalina issues (#3175)
- fix: Remove runtime panics in op dispatch (#3176, #3202, #3131)
- fix: BufReader.readString to actually return Deno.EOF at end (#3191)
- perf: faster TextDecoder (#3180, #3204)
- chore: Reenable std tests that were disabled during merge (#3159)
- chore: Remove old website (#3194, #3181)
- chore: Use windows-2019 image in Github Actions (#3198)
- chore: use v0.21.0 for subcommands (#3168)
- upgrade: V8 to 7.9.317.12 (#3208)

v0.21.0 / 2019.10.19

- feat: --reload flag to take arg for partial reload (#3109)
- feat: Allow "deno eval" to run code as module (#3148)
- feat: support --allow-net=:4500 (#3115)
- fix: Ensure DENO_DIR when saving the REPL history (#3106)
- fix: Update echo_server to new listen API (denoland/deno_std#625)
- fix: [prettier] deno fmt should format jsx/tsx files (#3118)
- fix: [tls] op_dial_tls is not registered and broken (#3121)
- fix: clearTimer bug (#3143)
- fix: remote jsx/tsx files were compiled as js/ts (#3125)
- perf: eager poll async ops in Isolate (#3046, #3128)
- chore: Move std/fs/path to std/path (#3100)
- upgrade: V8 to 7.9.304 (#3127)
- upgrade: prettier type definition (#3101)
- chore: Add debug build to github actions (#3127)
- chore: merge deno_std into deno repo (#3091, #3096)

v0.20.0 / 2019.10.06

In deno:

- feat: Add Deno.hostname() (#3032)
- feat: Add support for passing a key to Deno.env() (#2952)
- feat: JSX Support (#3038)
- feat: Replace Isolate::set_dispatch with Isolate::register_op (#3002, #3039, #3041)
- feat: window.onunload (#3023)
- fix: Async compiler processing (#3043)
- fix: Implement ignoreBOM option of UTF8Decoder in text_encoding (#3040)
- fix: Support top-level-await in TypeScript (#3024)
- fix: iterators on UrlSearchParams (#3044)
- fix: listenDefaults/dialDefaults may be overridden in some cases (#3027)
- upgrade: V8 to 7.9.218 (#3067)
- upgrade: rust to 1.38.0 (#3030)
- chore: Migrate CI to github actions (#3052, #3056, #3049, #3071, #3076, #3070, #3066, #3061, #3010)

- chore: Remove deno_cli_snapshots crate. Move //js to //cli/js (#3064)
- chore: use xeval from deno_std (#3058)

In deno_std:

- feat: test runner v2 (denoland/deno_std#604)
- feat: wss support with dialTLS (denoland/deno_std#615)
- fix(ws): mask must not be set by default for server (denoland/deno_std#616)
- fix: Implement expandGlob() and expandGlobSync() (denoland/deno_std#617)
- upgrade: eslint and @typescript-eslint (denoland/deno_std#621)

v0.19.0 / 2019.09.24

In deno:

- feat: Add Deno.dialTLS()
- feat: Make deno_cli installable via crates.io (#2946)
- feat: Remove test.py, use cargo test as test frontend (#2967)
- feat: dial/listen API change (#3000)
- feat: parallelize downloads from TS compiler (#2949)
- fix: Make `window` compatible with ts 3.6 (#2984)
- fix: Remove some non-standard web API constructors (#2970)
- fix: debug logging in runtime/compiler (#2953)
- fix: flag parsing of config file (#2996)
- fix: reschedule global timer if it fires earlier than expected (#2989)
- fix: type directive parsing (#2954)
- upgrade: V8 to 7.9.110 for top-level-await (#3015)
- upgrade: to TypeScript 3.6.3 (#2969)

In deno_std:

- feat: Implement BufReader.readString (denoland/deno_std#607)
- fix: TOML's key encoding (denoland/deno_std#612)
- fix: remove //testing/main.ts (denoland/deno_std#605)
- fix: types in example_client for ws module (denoland/deno_std#609)
- upgrade: mime-db to commit c50e0d1 (denoland/deno_std#608)

v0.18.0 / 2019.09.13

In deno:

- build: remove tools/build.py; cargo build is the build frontend now (#2865, #2874, #2876)
- feat: Make integration tests rust unit tests (#2884)
- feat: Set user agent for http client (#2916)
- feat: add bindings to run microtasks from Isolate (#2793)
- fix(fetch): implement bodyUsed (#2877)
- fix(url): basing in constructor (#2867, #2921)
- fix(xeval): incorrect chunk matching behavior (#2857)
- fix: Default 'this' to window in EventTarget (#2918)
- fix: Expose the DOM Body interface globally (#2903)
- fix: Keep all deno_std URLs in sync (#2930)
- fix: make 'deno fmt' faster (#2928)
- fix: panic during block_on (#2905)
- fix: panic during fetch (#2925)
- fix: path normalization in resolve_from_cwd() (#2875)

- fix: remove deprecated Deno.platform (#2895)
- fix: replace bad rid panics with errors (#2870)
- fix: type directives import (#2910)
- upgrade: V8 7.9.8 (#2907)
- upgrade: rust crates (#2937)

In deno_std:

- feat: Add xeval (denoland/deno_std#581)
- fix(flags): Parse builtin properties (denoland/deno_std#579)
- fix(uuid): Make it v4 rfc4122 compliant (denoland/deno_std#580)
- perf: Improve prettier speed by adding d.ts files (denoland/deno_std#591)
- upgrade: prettier to 1.18.2 (denoland/deno_std#592)

v0.17.0 / 2019.09.04

In deno:

- feat: Add window.queueMicrotask (#2844)
- feat: Support HTTP proxies in fetch (#2822)
- feat: Support `_` and `_error` in REPL (#2845, #2843)
- feat: add statusText for fetch (#2851)
- feat: implement Addr interface (#2821)
- fix: Improve error stacks for async ops (#2820)
- fix: add console.dirxml (#2835)
- fix: do not export `isConsoleInstance` (#2850)
- fix: set/clearTimeout's params should not be bigint (#2834, #2838)
- fix: shared queue requires aligned buffer (#2816)
- refactor: Remove Node build dependency and change how internal V8 snapshots are built (#2825, #2827, #2826, #2826)
- refactor: Remove flatbuffers (#2818, #2819, #2817, #2812, #2815, #2799)
- regression: Introduce regression in fetch's Request/Response stream API to support larger refactor (#2826)

In deno_std:

- fix: better paths handling in test runner (denoland/deno_std#574)
- fix: avoid prototype builtin `hasOwnProperty` (denoland/deno_std#577)
- fix: boolean regexp (denoland/deno_std#582)
- fix: printf should use padEnd and padStart (denoland/deno_std#583)
- fix: ws should use crypto getRandomValues (denoland/deno_std#584)

v0.16.0 / 2019.08.22

In deno:

- feat: "deno test" subcommand (#2783, #2784, #2800)
- feat: implement console.trace() (#2780)
- feat: support .d.ts files (#2746)
- feat: support custom inspection of objects (#2791)
- fix: dynamic import panic (#2792)
- fix: handle tsconfig.json with comments (#2773)
- fix: import map panics, use import map's location as its base URL (#2770)
- fix: set response.url (#2782)

In deno_std:

- feat: add overloaded form of unit test declaration (denoland/deno_std#563)
- feat: add printf implementation (fmt/sprintf.ts) (denoland/deno_std#566)
- feat: print out the failed tests after the summary (denoland/deno_std#554)
- feat: test runner (denoland/deno_std#516, denoland/deno_std#564, denoland/deno_std#568)
- fix: accept absolute root directories in the file server (denoland/deno_std#558)
- fix: refactor 'assertEquals' (denoland/deno_std#560)
- fix: test all text functions in colors module (denoland/deno_std#553)
- fix: move colors module into fmt module (denoland/deno_std#571)

v0.15.0 / 2019.08.13

In deno:

- feat: print cache location when no arg in deno info (#2752)
- fix: Dynamic import should respect permissions (#2764)
- fix: Propagate Url::to_file_path() errors instead of panicking (#2771)
- fix: cache paths on Windows are broken (#2760)
- fix: dynamic import base path problem for REPL and eval (#2757)
- fix: permission requirements for Deno.rename() and Deno.link() (#2737)

In deno_std: None

v0.14.0 / 2019.08.09

In deno:

- feat: remove `Deno.build.args` (#2728)
- feat: support native line ending conversion in the `Blob` constructor (#2695)
- feat: add option to delete the `Deno` namespace in a worker (#2717)
- feat: support starting workers using a blob: URL (#2729)
- feat: make `Deno.execPath()` a function (#2743, #2744)
- feat: support `await import(...)` syntax for dynamic module imports (#2516)
- fix: enforce permissions on `Deno.kill()`, `Deno.homeDir()` and `Deno.execPath()` (#2714, #2723)
- fix: `cargo build` now builds incrementally (#2740)
- fix: avoid REPL crash when DENO_DIR doesn't exist (#2727)
- fix: resolve worker module URLs relative to the host main module URL (#2751)
- doc: improve documentation on using the V8 profiler (#2742)

In deno_std:

- fix: make the 'ws' module (websockets) work again (denoland/deno_std#550)

v0.13.0 / 2019.07.31

In deno:

- feat: add debug info to ModuleResolutionError (#2697)
- feat: expose writeAll() and writeAllSync() (#2298)
- feat: Add --current-thread flag (#2702)
- fix: REPL shouldn't panic when it gets SIGINT (#2662)
- fix: Remap stack traces of unthrown errors (#2693)
- fix: bring back --no-fetch flag (#2671)
- fix: handle deno -v and deno --version (#2684)
- fix: make importmap flag global (#2687)

- fix: timer's params length (#2655)
- perf: Remove v8::Locker calls (#2665, #2664)

In deno_std:

- fix: Make shebangs Linux compatible (denoland/deno_std#545)
- fix: Ignore error of writing responses to aborted requests (denoland/deno_std#546)
- fix: use Deno.execPath where possible (denoland/deno_std#548)

v0.12.0 / 2019.07.16

In deno:

- feat: Support window.onload (#2643)
- feat: generate default file name for bundle when URL ends in a slash (#2625)
- fix: for '-' arg after script name (#2631)
- fix: upgrade v8 to 7.7.200 (#2624)

In deno_std:

- Rename catjson.ts to catj.ts (denoland/deno_std#533)
- Remove os.userHomeDir in favor of Deno.homeDir (denoland/deno_std#523)
- fix: emptydir on windows (denoland/deno_std#531)

v0.11.0 / 2019.07.06

In deno:

- feat: Add Deno.homeDir() (#2578)
- feat: Change Reader interface (#2591)
- feat: add bash completions (#2577)
- feat: parse CLI flags after script name (#2596)
- fix: multiple error messages for a missing file (#2587)
- fix: normalize Deno.execPath (#2598)
- fix: return useful error when import path has no ./ (#2605)
- fix: run blocking function on a different task (#2570)

In deno_std:

- feat: add UUID module (denoland/deno_std#479)
- feat: prettier support reading code from stdin (denoland/deno_std#498)

v0.10.0 / 2019.06.25

In deno:

- feat: improve module download progress (#2576)
- feat: improve 'deno install' (#2551)
- feat: log permission access with -L=info (#2518)
- feat: redirect process stdio to file (#2554)
- fix: add encodeInto to TextEncoder (#2558)
- fix: clearTimeout should convert to number (#2539)
- fix: clearTimeout.name / clearInterval.name (#2540)
- fix: event `isTrusted` is enumerable (#2543)
- fix: fetch() body now async iterable (#2563)
- fix: fetch() now handles redirects (#2561)
- fix: prevent multiple downloads of modules (#2477)

- fix: silent failure of WebAssembly.instantiate() (#2548)
- fix: urlSearchParams custom symbol iterator (#2537)

In deno_std

- feat(testing): Pretty output + Silent mode (denoland/deno_std#314)
- feat: Add os/userHomeDir (denoland/deno_std#521)
- feat: add catjson example (denoland/deno_std#517)
- feat: add encoding/hex module (denoland/deno_std#434)
- feat: improve installer (denoland/deno_std#512, denoland/deno_std#510, denoland/deno_std#499)
- fix: bundle/run handles Deno.args better. (denoland/deno_std#514)
- fix: file server should order filenames (denoland/deno_std#511)

v0.9.0 / 2019.06.15

In deno:

- feat: add deno install command (#2522)
- feat: URLSearchParams should work with custom iterator (#2512)
- feat: default output filename for deno bundle (#2484)
- feat: expose window.Response (#2515)
- feat: Add --seed for setting RNG seed (#2483)
- feat: Import maps (#2360)
- fix: setTimeout API adjustments (#2511, #2497)
- fix: URL and URLSearchParams bugs (#2495, #2488)
- fix: make global request type an interface (#2503)
- upgrade: V8 to 7.7.37 (#2492)

In deno_std:

- feat: installer (denoland/deno_std#489)
- feat: bundle loader (denoland/deno_std#480)

v0.8.0 / 2019.06.08

In deno:

- feat: Add 'bundle' subcommand. (#2467)
- feat: Handle compiler diagnostics in Rust (#2445)
- feat: add deno fmt --stdout option (#2439)
- feat: CLI defaults to run subcommand (#2451)
- fix: Compiler exit before emit if preEmitDiagnostics found (#2441)
- fix: Deno.core.evalContext & Deno.core.print (#2465)
- fix: Improve setup.py for package managers (#2423)
- fix: Use body when Request instance is passed to fetch (#2435)
- perf: Create fewer threads (#2476)
- upgrade: TypeScript to 3.5.1 (#2437)
- upgrade: std/prettier@0.5.0 to std/prettier@0.7.0 (#2425)

In deno_std:

- ci: Check file changes during test (denoland/deno_std#476)
- ci: Implement strict mode (denoland/deno_std#453)
- ci: Make CI config DRY (denoland/deno_std#470)
- encoding/csv: add easy api (denoland/deno_std#458)

- io: make port BufReader.readByte() return `number | EOF` (denoland/deno_std#472)
- ws: Add sec-websocket-version to handshake header (denoland/deno_std#468)

v0.7.0 / 2019.05.29

In deno:

- TS compiler refactor (#2380)
- add EventTarget implementation (#2377)
- add module and line no for Rust logger (#2409)
- re-fix permissions for dial and listen (#2400)
- Fix concurrent accepts (#2403)
- Rename --allow-high-precision to --allow-hrtime (#2398)
- Use tagged version of prettier in CLI (#2387)

In deno_std:

- io: refactor BufReader/Writer interfaces to be more idiomatic (denoland/deno_std#444)
- http: add rfc7230 handling (denoland/deno_std#451)
- http: add ParseHTTPVersion (denoland/deno_std#452)
- rename strings/strings.ts to strings/mod.ts (denoland/deno_std#449)
- Prettier: support for specified files and glob mode (denoland/deno_std#438)
- Add encoding/csv (denoland/deno_std#432)
- rename bytes/bytes.ts to bytes/mod.ts
- remove function prefix of bytes module
- add bytes.repeat() (denoland/deno_std#446)
- http: fix content-length checking (denoland/deno_std#437)
- Added isGlob function (denoland/deno_std#433)
- http: send an empty response body if none is provided (denoland/deno_std#429)
- http: make server handle bad client requests properly (denoland/deno_std#419)
- fix(fileserver): wrong url href of displayed files (denoland/deno_std#426)
- http: delete conn parameter in readRequest (denoland/deno_std#430)
- Rename //multipart/multipart.ts to //mime/multipart.ts (denoland/deno_std#420)
- feat(prettier): output to stdout instead of write file by default unless specified --write flag (denoland/deno_std#332)

v0.6.0 / 2019.05.20

In deno:

- Fix permissions for dial and listen (#2373)
- Add crypto.getRandomValues() (#2327)
- Don't print new line if progress bar was not used (#2374)
- Remove FileInfo.path (#2313)

In deno_std

- Clean up HTTP async iterator code (denoland/deno_std#411)
- fix: add esnext lib to tsconfig.json (denoland/deno_std#416)
- feat(fs): add copy/copySync (denoland/deno_std#278)
- feat: add Tar and Untar classes (denoland/deno_std#388)
- ws: make acceptable() more robust (denoland/deno_std#404)

v0.5.0 / 2019.05.11

In deno:

- Add progress bar (#2309)
- fix: edge case in toAsyncIterator (#2335)
- Upgrade rust crates (#2334)
- white listed permissions (#2129 #2317)
- Add Deno.chown (#2292)

In deno_std:

- benching: use performance.now (denoland/deno_std#385)
- bytes fix bytesFindIndex and bytesFindLastIndex (denoland/deno_std#381)

v0.4.0 / 2019.05.03

In deno:

- add "deno run" subcommand (#2215)
- add "deno xeval" subcommand (#2260)
- add --no-fetch CLI flag to prevent remote downloads (#2213)
- Fix: deno --v8-options does not print v8 options (#2277)
- Performance improvements and fix memory leaks (#2259, #2238)
- Add Request global constructor (#2253)
- fs: add Deno.utime/Deno.utimeSync (#2241)
- Make `atob` follow the spec (#2242)
- Upgrade V8 to 7.6.53 (#2236)
- Remove ? from URL when deleting all params (#2217)
- Add support for custom tsconfig.json (#2089)
- URLSearchParams init with itself (#2218)

In deno_std:

- textproto: fix invalid header error and move tests (#369)
- Add http/cookie improvements (#368, #359)
- fix ensureLink (#360)

v0.3.10 / 2019.04.25

In deno:

- Fix "deno types" (#2209)
- CLI flags/subcommand rearrangement (#2210, #2212)

v0.3.9 / 2019.04.25

In deno:

- Fix #2033, shared queue push bug (#2158)
- Fix panic handler (#2188)
- cli: Change "deno --types" to "deno types" and "deno --prefetch" to "deno prefetch" (#2157)
- Make Deno/Deno.core not deletable/writable (#2153)
- Add Deno.kill(pid, signo) and process.kill(signo) (Unix only) (#2177)
- symlink: Ignore type parameter on non-Windows platforms (#2185)
- upgrade rust crates (#2186)
- core: make Isolate concrete, remove Dispatch trait (#2183)

In deno_std:

- http: add cookie module (#338)
- fs: add getFileInfoType() (#341)
- fs: add ensureLink/ensureLinkSync (#353)
- fs: add ensureSymlink/ensureSymlinkSync (#268)
- fs: add readFileStr, writeFileStr (#276, #340)
- testing: support Sets in asserts.equals (#350)

v0.3.8 / 2019.04.19

In deno:

- Async module loading (#2084 #2133)
- core: improve tail latency (#2131)
- third_party: upgrade rust crates
- add custom panic handler to avoid silent failures (#2098)
- fix absolute path resolution from remote (#2109)
- Add deno eval subcommand (#2102)
- fix: re-expose DomFile (#2100)
- avoid prototype builtin hasOwnProperty (#2144)

In deno_std:

- Enforce HTTP/1.1 pipeline response order (deno_std#331)
- EOL add mixed detection (deno_std#325)
- Added read file str (deno_std#276)
- add writeFileStr and update documentation (deno_std#340)

v0.3.7 / 2019.04.11

In deno:

- Use clap for command line flag parsing (#2093, #2068, #2065, #2025)
- Allow high precision performance.now() (#1977)
- Fix `console instanceof Console` (#2073)
- Add link/linkSync fs call for hardlinks (#2074)
- build: Use -O3 instead of -O (#2070)

In deno_std:

- fs: add fs/mod.ts entry point (deno_std#272)
- prettier: change flag parsing (deno_std#327)
- fs: add EOL detect / format (deno_std#289)
- fs: ensure exists file/dir must be the same type or it will throw error (deno_std#294)

v0.3.6 / 2019.04.04

In deno:

- upgrade rust crates (#2016)
- EventTarget improvements (#2019, #2018)
- Upgrade to TypeScript 3.4.1 (#2027)
- console/toString improvements (#2032, #2042, #2041, #2040)
- Add web worker JS API (#1993, #2039)
- Fix redirect module resolution bug (#2031)
- core: publish to crates.io (#2015,#2022, #2023, #2024)
- core: add RecursiveLoad for async module loading (#2034)

In deno_std:

- toml: Full support of inline table (deno_std#320)
- fix benchmarks not returning on deno 0.3.4+ (deno_std#317)

v0.3.5 / 2019.03.28

In deno:

- Add Process.stderrOutput() (#1828)
- Check params in Event and CustomEvent (#2011, #1997)
- Merge --reload and --recompile flags (#2003)
- Add Deno.openSync, .readSync, .writeSync, .seekSync (#2000)
- Do not close file on invalid seek mode (#2004)
- Fix bug when shared queue is overflowed (#1992)
- core: Resolve callback moved from Behavior to mod_instantiate() (#1999)
- core: libdeno and DenoCore renamed to Deno.core (#1998)
- core: Allow terminating an Isolate from another thread (#1982)

In deno_std:

- Add TOML parsing module (#300)
- testing: turn off exitOnFail by default (#307, #309)
- Fix assertEquals for RegExp & Date (#305)
- Fix prettier check in empty files (#302)
- remove unnecessary path.resolve in move/readJson/writeJson (#292)
- fix: fs.exists not work for symlink (#291)
- Add prettier styling options (#281)

v0.3.4 / 2019.03.20

In deno itself:

- Performance improvements (#1959, #1938)
- Improve pretty printing of objects (#1969)
- More permissions prompt options (#1926)

In deno_std:

- Add prettier styling options (#281)
- Extract internal method isSubdir to fs/utils.ts (#285)
- Add strings/pad (#282)

v0.3.3 / 2019.03.13

In deno itself:

- Rename Deno.build.gnArgs to Deno.build.args (#1912, #1909)
- Upgrade to TypeScript 3.3 (#1908)
- Basic Arm64 support (#1887)
- Remove builtin "deno" module, use Deno global var (#1895)
- Improvements to internal deno_core crate (#1904, #1914)
- Add --no-prompt flag for non-interactive environments (#1913)

In deno_std

- Add fs extras: ensureDir, ensureFile, readJson, emptyDir, move, exists (#269, #266, #264, #263, #260)

- Datetime module improvement (#259)
- asserts: Add unimplemented, unreachable, assertNotEquals, assertArrayContains (#246, #248)

v0.3.2 / 2019.03.06

In deno itself:

- Reorganize version and platform into Deno.build and Deno.version (#1879)
- Allow inspection and revocation of permissions (#1875)
- Fix unicode output on Windows (#1876)
- Add Deno.build.gnArgs (#1845)
- Fix security bug #1858 (#1864, #1874)
- Replace deno.land/x/std links with deno.land/std/ (#1890)

In deno_std:

- Move asserts out of testing/mod.ts into testing/assert.ts Rename assertEqual to assertEquals (#240, #242)
- Update mime-db to 1.38.0 (#238)
- Use pretty assertEqual in testing (#234)
- Add eslint to CI (#235)
- Refactor WebSockets (#173)
- Allow for parallel testing (#224)
- testing: use color module for displaying colors (#223)
- Glob integration for the FS walker (#219)

v0.3.1 / 2019.02.27

- Add import.meta.main (#1835)
- Fix console.table display of Map (#1839)
- New low-level Rust API (#1827)
- Upgrade V8 to 7.4.238 (#1849)
- Upgrade crates (#1848)

v0.3.0 / 2019.02.18

The major API change in this release is that instead of importing a `"deno"` module, there is now a global variable called `Deno`. This allows code that does deno-specific stuff to still operate in browsers. We will remain backward compatible with the old way of importing core functionality, but it will be removed in the near future, so please update your code. See #1748 for more details.

- Add Deno global namespace object (#1748)
- Add window.location (#1761)
- Add back typescript version number and add Deno.version object (#1788)
- Add `seek` and implement `Seeker` on `File` (#1797)
- Add Deno.execPath (#1743)
- Fix behavior for extensionless files with .mime file (#1779)
- Add env option in Deno.run (#1773)
- Turn on `v8_postmortem_support` (#1758)
- Upgrade V8 to 7.4.158 (#1767)
- Use proper directory for cache files (#1763)
- REPL multiline support with recoverable errors (#1731)
- Respect `NO_COLOR` in TypeScript output (#1736)
- Support scoped variables, unblock REPL async op, and REPL error colors (#1721)

v0.2.11 / 2019.02.08

- Add deps to --info output (#1720)
- Add --allow-read (#1689)
- Add deno.isTTY() (#1622)
- Add emojis to permission prompts (#1684)
- Add basic WebAssembly support (#1677)
- Add `NO_COLOR` support https://no-color.org/ (#1716)
- Add color exceptions (#1698)
- Fix: do not load cache files when recompile flag is set (#1695)
- Upgrade V8 to 7.4.98 (#1640)

v0.2.10 / 2019.02.02

- Add --fmt (#1646)
- Add --info (#1647, #1660)
- Better error message for bad filename CLI argument. (#1650)
- Clarify writeFile options and avoid unexpected perm modification (#1643)
- Add performance.now (#1633)
- Add import.meta.url (#1624)

v0.2.9 / 2019.01.29

- Add REPL functions "help" and "exit" (#1563)
- Split out compiler snapshot (#1566)
- Combine deno.removeAll into deno.remove (#1596)
- Add console.table (#1608)
- Add console.clear() (#1562)
- console output with format (#1565)
- env key/value should both be strings (#1567)
- Add CustomEvent API (#1505)

v0.2.8 / 2019.01.19

- Add --prefetch flag for deps prefetch without running (#1475)
- Kill all pending accepts when TCP listener is closed (#1517)
- Add globalThis definition to runtime (#1534)
- mkdir should not be recursive by default (#1530)
- Avoid crashes on ES module resolution when module not found (#1546)

v0.2.7 / 2019.01.14

- Use rust 2018 edition
- Native ES modules (#1460 #1492 #1512 #1514)
- Properly parse network addresses (#1515)
- Added rid to Conn interface (#1513)
- Prevent segfault when eval throws an error (#1411)
- Add --allow-all flag (#1482)

v0.2.6 / 2019.01.06

- Implement console.groupCollapsed (#1452)
- Add deno.pid (#1464)
- Add Event web API (#1059)
- Support more fetch init body types (#1449)

v0.2.5 / 2018.12.31

- Runtime argument checks (#1427 #1415)
- Lazily create .mime files only with mismatch/no extension (#1417)
- Fix FormData.name (#1412)
- Print string with NULL '\0' (#1428)

v0.2.4 / 2018.12.23

- "cargo build" support (#1369 #1296 #1377 #1379)
- Remove support for extensionless import (#1396)
- Upgrade V8 to 7.2.502.16 (#1403)
- make stdout unbuffered (#1355)
- Implement `Body.formData` for fetch (#1393)
- Improve handling of non-coercable objects in assertEqual (#1385)
- Avoid fetch segfault on empty Uri (#1394)
- Expose deno.inspect (#1378)
- Add illegal header name and value guards (#1375)
- Fix URLSearchParams set() and constructor() (#1368)
- Remove prebuilt v8 support (#1369)
- Enable jumbo build in release. (#1362)
- Add URL implementation (#1359)
- Add console.count and console.time (#1358)
- runtime arg check `URLSearchParams` (#1390)

v0.2.3 / 2018.12.14

- console.assert should not throw error (#1335)
- Support more modes in deno.open (#1282, #1336)
- Simplify code fetch logic (#1322)
- readDir entry mode (#1326)
- Use stderr for exceptions (#1303)
- console.log formatting improvements (#1327, #1299)
- Expose TooLarge error code for buffers (#1298)

v0.2.2 / 2018.12.07

- Don't crash when .mime file not exist in cache (#1291)
- Process source maps in Rust instead of JS (#1280)
- Use alternate TextEncoder/TextDecoder implementation (#1281)
- Upgrade flatbuffers to 80d148
- Fix memory leaks (#1265, #1275)

v0.2.1 / 2018.11.30

- Allow async functions in REPL (#1233)
- Handle Location header relative URI (#1240)
- Add deno.readAll() (#1234)
- Add Process.output (#1235)
- Upgrade to TypeScript 3.2.1
- Upgrade crates: tokio 0.1.13, hyper 0.12.16, ring 0.13.5

v0.2.0 / 2018.11.27 / Mildly usable

[An intro talk was recorded.](#)

Stability and usability improvements. `fetch()` is 90% functional now. Basic REPL support was added. Shebang support was added. Command-line argument parsing was improved. A forwarding service `https://deno.land/x` was set up for Deno code. Example code has been posted to deno.land/x/examples and deno.land/x/net.

The resources table was added to abstract various types of I/O streams and other allocated state. A resource is an integer identifier which maps to some Rust object. It can be used with various ops, particularly read and write.

Changes since v0.1.12:

- First pass at running subprocesses (#1156)
- Improve flag parsing (#1200)
- Improve fetch() (#1194 #1188 #1102)
- Support shebang (#1197)

v0.1.12 / 2018.11.12

- Update to TypeScript 3.1.6 (#1177)
- Fixes Headers type not available. (#1175)
- Reader/Writer to use Uint8Array not ArrayBufferView (#1171)
- Fixes importing modules starting with 'http'. (#1167)
- build: Use target/ instead of out/ (#1153)
- Support repl multiline input (#1165)

v0.1.11 / 2018.11.05

- Performance and stability improvements on all platforms.
- Add repl (#998)
- Add deno.Buffer (#1121)
- Support cargo check (#1128)
- Upgrade Rust crates and Flatbuffers. (#1145, #1127)
- Add helper to turn deno.Reader into async iterator (#1130)
- Add ability to load JSON as modules (#1065)
- Add deno.resources() (#1119)
- Add application/x-typescript mime type support (#1111)

v0.1.10 / 2018.10.27

- Add URLSearchParams (#1049)
- Implement clone for FetchResponse (#1054)
- Use content-type headers when importing from URLs. (#1020)
- Use checkJs option, JavaScript will be type checked and users can supply JSDoc type annotations that will be enforced by Deno (#1068)
- Add separate http/https cache dirs to DENO_DIR (#971)
- Support https in fetch. (#1100)
- Add chmod/chmodSync on unix (#1088)
- Remove broken features: --deps and trace() (#1103)
- Ergonomics: Prompt TTY for permission escalation (#1081)

v0.1.9 / 2018.10.20

- Performance and stability improvements on all platforms.
- Add cwd() and chdir() #907
- Specify deno_dir location with env var DENO_DIR #970
- Make fetch() header compliant with the current spec #1019

- Upgrade TypeScript to 3.1.3
- Upgrade V8 to 7.1.302.4

v0.1.8 / 2018.10.12 / Connecting to Tokio / Fleshing out APIs

Most file system ops were implemented. Basic TCP networking is implemented. Basic stdio streams exposed. And many random OS facilities were exposed (e.g. environmental variables)

Tokio was chosen as the backing event loop library. A careful mapping of JS Promises onto Rust Futures was made, preserving error handling and the ability to execute synchronously in the main thread.

Continuous benchmarks were added: https://denoland.github.io/deno/ Performance issues are beginning to be addressed.

"deno --types" was added to reference runtime APIs.

Working towards https://github.com/denoland/deno/milestone/2 We expect v0.2 to be released in last October or early November.

Changes since v0.1.7:

- Fix promise reject issue (#936)
- Add --types command line flag.
- Add metrics()
- Add redirect follow feature #934
- Fix clearTimer bug #942
- Improve error printing #935
- Expose I/O interfaces Closer, Seeker, ReaderCloser, WriteCloser, ReadSeeker, WriteSeeker, ReadWriteCloser, ReadWriteSeeker
- Fix silent death on double await #919
- Add Conn.closeRead() and Conn.closeWrite() #903

v0.1.7 / 2018.10.04

- Improve fetch headers (#853)
- Add deno.truncate (#805)
- Add copyFile/copyFileSync (#863)
- Limit depth of output in console.log for nested objects, and add console.dir (#826)
- Guess extensions on extension not provided (#859)
- Renames: deno.platform -> deno.platform.os deno.arch -> deno.platform.arch
- Upgrade TS to 3.0.3
- Add readDirSync(), readDir()
- Add support for TCP servers and clients. (#884) Adds deno.listen(), deno.dial(), deno.Listener and deno.Conn.

v0.1.6 / 2018.09.28

- Adds deno.stdin, deno.stdout, deno.stderr, deno.open(), deno.write(), deno.read(), deno.Reader, deno.Writer, deno.copy() #846
- Print 'Compiling' when compiling TS.
- Support zero-copy for writeFile() writeFileSync() #838
- Fixes eval error bug #837
- Make Deno multithreaded #782
- console.warn() goes to stderr #810
- Add deno.readlink()/readlinkSync() #797

- Add --recompile flag #801
- Use constructor.name to print out function type #664
- Rename deno.argv to deno.args
- Add deno.trace() #795
- Continuous benchmarks

v0.1.5 / 2018.09.21

- Add atob() btoa() #776
- Add deno.arch deno.platform #773
- Add deno.symlink() and deno.symlinkSync() #742
- Add deno.mkdir() and deno.mkdirSync() #746
- Add deno.makeTempDir() #740
- Improvements to FileInfo interface #765, #761
- Add fetch.blob()
- Upgrade V8 to 7.0.276.15
- Upgrade Rust crates

v0.1.4 / 2018.09.12

- Support headers in fetch()
- Adds many async fs functions: deno.rename() deno.remove(), deno.removeAll(), deno.removeSync(), deno.removeAllSync(), deno.mkdir(), deno.stat(), deno.lstat() deno.readFile() and deno.writeFile().
- Add mode in FileInfo
- Access error codes via error.kind
- Check --allow-net permissions when using fetch()
- Add deno --deps for listing deps of a script.

v0.1.3 / 2018.09.05 / Scale binding infrastructure

ETA v.0.2 October 2018 https://github.com/denoland/deno/milestone/2

We decided to use Tokio https://tokio.rs/ to provide asynchronous I/O, thread pool execution, and as a base for high level support for various internet protocols like HTTP. Tokio is strongly designed around the idea of Futures - which map quite well onto JavaScript promises. We want to make it as easy as possible to start a Tokio future from JavaScript and get a Promise for handling it. We expect this to result in preliminary file system operations, fetch() for http. Additionally we are working on CI, release, and benchmarking infrastructure to scale development.

Changes since v0.1.2:

- Fixes module resolution error #645
- Better flag parsing
- lStatSync -> lstatSync
- Added deno.renameSync()
- Added deno.mkdirSync()
- Fix circular dependencies #653
- Added deno.env() and --allow-env

v0.1.2 / 2018.08.30

- Added https import support.
- Added deno.makeTempDirSync().
- Added deno.lstatSync() and deno.statSync().

v0.1.1 / 2018.08.27

v0.1.0 / 2018.08.23 / Rust rewrite and V8 snapshot

Complete! https://github.com/denoland/deno/milestone/1

Go is a garbage collected language and we are worried that combining it with V8's GC will lead to difficult contention problems down the road.

The V8Worker2 binding/concept is being ported to a new C++ library called libdeno. libdeno will include the entire JS runtime as a V8 snapshot. It still follows the message passing paradigm. Rust will be bound to this library to implement the privileged part of deno. See deno2/README.md for more details.

V8 Snapshots allow deno to avoid recompiling the TypeScript compiler at startup. This is already working.

When the rewrite is at feature parity with the Go prototype, we will release binaries for people to try.

v0.0.0 / 2018.05.14 - 2018.06.22 / Golang Prototype

https://github.com/denoland/deno/tree/golang

https://www.youtube.com/watch?v=M3BM9TB-8yA

https://tinyclouds.org/jsconf2018.pdf

2007-2017 / Prehistory

https://github.com/ry/v8worker

https://libuv.org/

https://tinyclouds.org/iocp-links.html

https://nodejs.org/

https://github.com/nodejs/http-parser

https://tinyclouds.org/libebb/