

changelog.md for @push.rocks/smartrust

2026-03-14 - 1.3.2 - fix(rustbinarylocator)

support resolving platform-suffixed local Rust binaries

- Checks local Rust build paths for binaries with platform suffixes such as `_linux_amd64` in addition to unsuffixed names
- Adds platform and architecture suffix mapping for linux, darwin, windows, x64, and arm64

2026-02-26 - 1.3.1 - fix(readme)

document socket transport and clarify stdio/socket differences in README

- Add 'Two Transport Modes' section documenting stdio (spawn) and socket (connect) modes
- Add examples for `connect()`, socket usage, and auto-reconnect with exponential backoff
- Clarify protocol is transport-agnostic and update ready/stream/event descriptions
- Update event docs: mark `stderr` as stdio-only and add 'reconnected' event for socket transports
- Clarify `kill()` behavior for both stdio and socket transports
- Add API reference entries for `SocketTransport`, `StdioTransport`, `ISocketConnectOptions`, `IRustTransport`, and `LineScanner`
- Add platform notes, architecture diagram, and minimal Rust/socket usage guidance

2026-02-26 - 1.3.0 - feat(transport)

introduce transport abstraction and socket-mode support for RustBridge

- Add IRustTransport interface and two transport implementations: StdioTransport (spawns child process and uses stdin/stdout) and SocketTransport (connects to Unix socket / Windows named pipe).
- Refactor RustBridge to use a transport abstraction (connectWithTransport) and add connect(socketPath) to attach to an existing daemon via socket.
- Introduce LineScanner: a buffer-based newline scanner used by both transports to handle large/newline-delimited messages and avoid OOMs.
- Add socket connection options (autoReconnect, reconnectBaseDelayMs, reconnectMaxDelayMs, maxReconnectAttempts) and implement auto-reconnect/backoff behavior in SocketTransport.
- Implement backpressure-aware write semantics and proper disconnect/cleanup for both transports; RustBridge.kill() now disconnects the transport instead of directly managing processes.
- Add tests and tooling: socket transport tests, line scanner tests, and a mock-socket-server.mjs helper script for testing socket mode.
- Export new symbols (StdioTransport, SocketTransport, LineScanner) and update plugins to expose net; update interfaces to export transport types.

2026-02-12 - 1.2.1 - fix(rust-binary-locator)

auto-fix missing execute permission for located Rust binaries

- If a located binary exists but lacks the execute bit, attempt to chmod it to 0o755 and treat it as executable.
- Logs an info message when the auto-fix is applied: 'Auto-fixed missing execute permission on: <filePath>'.
- Addresses cases where npm/pnpm installs remove the execute permission from bundled binaries.

2026-02-11 - 1.2.0 - feat(rustbridge)

add streaming responses and robust large-payload/backpressure handling to RustBridge

- Introduce StreamingResponse type and export it (for-await-of iterator + .result promise)

- Add `sendCommandStreaming` API to send streaming commands and receive chunks + final result
- Implement buffer-based stdout newline scanner to handle large messages and avoid readline limits
- Add backpressure-aware `writeToStdin` to wait for drain when writing large outbound payloads
- Add `maxPayloadSize` option and enforce outbound/inbound size checks to prevent OOMs
- Add `streamTimeoutMs` (inactivity timeout) and reset timeout on each received chunk
- Improve stderr handling (cross-chunk buffering and trimmed emits)
- Update mock test binary and extensive tests for streaming, large payloads, concurrency, and error cases
- Add TypeScript types for streaming commands (`TStreamingCommandKeys`, `TExtractChunk`, `IManagementStreamChunk`)

2026-02-10 - 1.1.2 - fix(rust-binary-locator)

use `import.meta.resolve` and `url.fileURLToPath` to locate bundled Rust binary in ESM environments

- Replace `require.resolve` with `import.meta.resolve` to support ESM module resolution
- Convert resolved file URL to a filesystem path using `url.fileURLToPath`
- Export the `url` module from `ts/plugins` to provide `fileURLToPath`

2026-02-10 - 1.1.1 - fix(readme)

update README with comprehensive documentation, usage examples, API reference, installation instructions, and legal/company information

- Rewrote `readme.md` ($\approx +298 -3$ lines) to add detailed install, overview, IPC protocol, command definition examples, usage, API reference, issue reporting & security guidance, and legal/trademark/company information.
- Documentation-only change — no source code modified.
- Current package version is 1.1.0; recommend a patch release

2026-02-10 - 1.1.0 - feat(rustbridge)

add RustBridge and RustBinaryLocator with typed IPC interfaces, plugins, tests and mock runner; export from index; add npm registries

- Introduce RustBridge: spawn and manage a child binary, JSON-over-stdin/stdout request/response handling, events, timeouts, pending request tracking, kill/cleanup logic.
- Introduce RustBinaryLocator: multi-strategy binary discovery (explicit path, env var, platform-specific package, local build paths, system PATH) with caching and logger hooks.
- Add IPC and config TypeScript interfaces (IManagementRequest/Response/Event, ICommandDefinition, IBinaryLocatorOptions, IRustBridgeOptions) and re-export via interfaces/index.ts.
- Update ts/plugins.ts to export fs, child_process, readline and events for easier native integration.
- Add tests for RustBridge and RustBinaryLocator plus a test helper mock-rust-binary.mjs to simulate the IPC protocol and exercise commands, events, timeouts and locator behaviors.
- Update ts/index.ts to export RustBridge and RustBinaryLocator and export interfaces; update npmextra.json to include internal Verdaccio registry alongside npmjs.org.

2026-02-08 - 1.0.2 - fix()

no changes

- No changes detected in git diff; no release necessary.

2026-02-08 - 1.0.1 - initial release

Initial release of the project.

- Initial commit creating the project repository and baseline files.
- Tagged as version 1.0.1.

Revision #2

Created 2026-03-28 13:11:37 UTC by foss.global Team

Updated 2026-03-29 16:54:08 UTC by foss.global Team