changelog.md for @push.rocks/smartfs

2026-03-06 - 1.5.0 - feat(rust-provider)

add cross-runtime Rust provider tests and docs; simplify bridge event handling and bump tstest

- Added comprehensive Rust provider test file supporting Node, Bun and Deno (test/test.rust.provider.node+bun+deno.ts) and removed the older node+bun-only test
- Simplified Rust provider bridge startup by removing a custom .on() override and relying on the bridge's inherited EventEmitter behavior
- Updated readme with new features: Directory Copy & Move, Multi-Runtime support, and expanded exported type list in examples
- Bumped dev dependency @git.zone/tstest from ^3.2.0 to ^3.3.0

2026-03-05 - 1.4.0 - feat(rust-provider)

Add Rust-backed provider with XFS-safe durability via IPC bridge, TypeScript provider, tests and docs

- Add Rust workspace and crates (smartfs-protocol, smartfs-core, smartfs-bin) with Cargo.toml and Cargo.lock
- Implement filesystem operations in Rust with XFS-safe parent fsyncs, streaming, watch support and IPC protocol types (smartfs-protocol)
- Add Rust binary (smartfs-bin) implementing management/IPC mode and core ops, plus watch manager and write-stream handling
- Add TypeScript bridge/provider (ts/providers/smartfs.provider.rust.ts), export provider from ts/index.ts, and include @push.rocks/smartrust in plugins

- Add integration tests for the Rust provider (test/test.rust.provider.node+bun.ts)
- Update packaging and tooling: package.json scripts and devDependencies (tsrust added/updated), npmextra.json target entry, .gitignore rust/target, and README updates

2026-03-05 - 1.3.3 - fix(smartfs.provider.node)

replace synchronous readdirSync with async await fs.readdir for directory listings in the Node provider to avoid blocking the event loop

- Replaced fsSync.readdirSync with await fs.readdir in listDirectory and listDirectoryRecursive.
- Switches from a blocking filesystem call to the non-blocking Node fs API in the node provider.
- Patch bump from 1.3.2 to 1.3.3 is recommended.

2026-03-05 - 1.3.2 - fix(provider(node))

use synchronous readdir to avoid partial results on some filesystems (e.g., XFS) when the process receives signals

- Replaced async fs.readdir with fsSync.readdirSync in ts/providers/smartfs.provider.node.ts
- Added comments explaining that async readdir can return partial results on XFS/mounted filesystems when the process receives signals; synchronous readdirSync completes the getdents64 syscall without event-loop interruption

2025-12-16 - 1.3.1 - fix(docs)

docs(readme): add "Directory Copy & Move" section with examples and options

- Adds README documentation for recursive directory copy and move with usage examples (basic copy/move, copy with filter, overwrite, preserve timestamps, applyFilter).
- Documents conflict handling modes for copy/move: merge (default), error, and replace.

- Documentation-only change — no code or API changes; recommended patch version bump.

2025-12-16 - 1.3.0 - feat(smartfs.directory)

feat(smartfs.directory): add directory copy/move with conflict handling and options

- Implement Directory.copy(targetPath) and Directory.move(targetPath) with provider-backed file operations (createDirectory, listDirectory, copyFile, deleteDirectory).
- Add new directory options and fluent setters: applyFilter, overwrite, preserveTimestamps, onConflict (defaults: applyFilter=true, overwrite=false, preserveTimestamps=false, onConflict='merge').
- Copy supports recursive listing, optional filtering (applyFilter), overwrite behavior and timestamp preservation; onConflict supports 'merge'|'error'|'replace'. Move performs copy then deletes the source.
- Add comprehensive tests for copy/move: basic copy, recursive copy, filter-based copy, applyFilter(false) behavior, overwrite handling, onConflict error/replace cases, move semantics, and copying empty directories.
- Update npmextra.json to use scoped keys (@git.zone/cli, @ship.zone/szci) and add release registry/access configuration.

2025-12-02 - 1.2.0 - feat(smartfs.directory)

Add directory treeHash: deterministic content-based hashing of directory trees with streaming and algorithm option

- Implement treeHash(options?) on SmartFsDirectory which computes a deterministic hash of a directory tree by hashing relative file paths and streaming file contents (default algorithm: 'sha256').
- Introduce ITreeHashOptions type (algorithm?: string) to allow selecting the hash algorithm (e.g. 'sha256', 'sha512').
- Use Node.js crypto to update the hash incrementally while streaming file data to keep memory usage low.
- Add tests in test/test.node.provider.ts covering treeHash behavior, determinism, algorithm selection, and empty-directory hashing.

- Update README with documentation, examples and explanation of treeHash use cases and behavior.

2025-11-30 - 1.1.3 - fix(smartfs.provider.node)

Default createDirectory to recursive=true when option not provided in Node provider

- Node provider: createDirectory now defaults to recursive=true when options.recursive is undefined.
- Prevents errors when creating nested directories without explicitly passing the recursive option.
- No API signature changes; behavior change is limited to the Node provider implementation.

2025-11-29 - 1.1.2 - fix(SmartFsProviderNode)

Fix Node provider watch path handling and remove main test entry

- Node provider: detect at start whether the watched path is a file or directory (fs.stat) and build fullPath accordingly so watching a single file does not incorrectly join the filename onto the file path.
- Watch callback: ensure events are evaluated against the configured filter using the correct full path.
- Tests: removed test/test.ts (main test entry that previously imported provider test files).

2025-11-29 - 1.1.1 - fix(smartfs.provider.node)

Default deleteDirectory to recursive=true in Node provider

- Changed SmartFsProviderNode.deleteDirectory to use recursive: options?.recursive ?? true when calling fs.rm.
- Directories will now be removed recursively by default when no recursive option is provided (was previously undefined).
- Retains force: true behavior to ignore missing targets and suppress errors.

2025-11-21 - 1.1.0 - feat(core)

Add SmartFS core library with providers, builders, interfaces, docs, tests and CI

- Add core TypeScript sources and public exports: SmartFs, SmartFsFile, SmartFsDirectory, SmartFsTransaction, SmartFsWatcher and ts/index.ts
- Add two providers: SmartFsProviderNode (Node.js fs/promises + fs.watch) and SmartFsProviderMemory (in-memory implementation used for testing)
- Add provider and type contracts: ISmartFsProvider, IProviderCapabilities and comprehensive mod.types definitions
- Implement transactions with prepare/execute/rollback, atomic writes, Web Streams-based read/write streams, and file watching with debouncing and filters
- Add tests entry (test/test.ts) and test scaffolding for memory and node providers
- Add package configuration (package.json, tsconfig.json, npmextra.json), documentation (readme.md, readme.hints.md) and plugins/paths helpers
- Add CI workflows and .gitignore

2025-11-21 - 1.0.1 - initial release

Initial project commit and setup.

- Project initialized with the initial scaffold and files
- Basic project configuration and versioning (1.0.1)

Revision #2
Created 2026-03-28 13:11:34 UTC by foss.global Team
Updated 2026-03-29 16:54:05 UTC by foss.global Team