

changelog.md for @push.rocks/smartarchive

2026-01-01 - 5.2.0 - feat(tartools)

add streaming TAR support (tar-stream), Node.js streaming APIs for TarTools, and browser / web bundle docs

- Add tar-stream runtime dependency and @types/tar-stream devDependency
- Introduce streaming TarTools APIs: getPackStream, addFileToPack, getExtractStream, extractToDirectory, getDirectoryPackStream, getDirectoryPackStreamGz
- Switch SmartArchive TAR extraction to use tar-stream extract for true streaming ingestion of entries
- Export tarStream in plugins and export ITarPackFileOptions from the Node.js entrypoint
- Update packDirectory/packDirectoryToTarGz to handle files safely and use fflate.gzipSync for buffer-based gzipping
- README updates: document /web browser bundle, browser usage examples, Uint8Array guidance, updated feature table and streaming examples

2026-01-01 - 5.1.0 - feat(archive)

introduce ts_shared browser-compatible layer, refactor Node-specific tools to wrap/shared implementations, and modernize archive handling

- Split code into ts_shared (browser-compatible) and ts_web entrypoint; node-specific wrappers remain under ts/
- Switched TAR implementation from tar-stream to modern-tar and replaced stream-based TAR handling with Uint8Array-based pack/unpack
- Normalized shared APIs to use Uint8Array (instead of Buffer) for browser compatibility — callers may need to adapt Buffer/Uint8Array usage (breaking)
- Moved BZIP2, GZIP, ZIP, TAR logic into ts_shared and updated plugins to re-export shared plugins for web builds
- Adjusted classes.smartarchive to consume shared tools and convert between Buffer and Uint8Array where needed

- Added package.json exports for "." and "./web", bumped several dependency/devDependency versions, and added modern-tar and related changes
- Updated npmextra.json with new scoped configuration and release registries
- Removed pnpm-workspace.yaml entries and deleted several legacy Node-only files in favor of shared implementations

2025-11-25 - 5.0.1 - fix(ziptools,gziptools)

Use fflate synchronous APIs for ZIP and GZIP operations for Deno compatibility; add TEntryFilter type and small docs/tests cleanup

- Replace fflate async APIs (zip, unzip, gzip, gunzip with callbacks) with synchronous counterparts (zipSync, unzipSync, gzipSync, gunzipSync) to avoid Web Worker issues in Deno
- ZipCompressionStream.finalize now uses fflate.zipSync and emits compressed Buffer synchronously
- GzipTools.compress / decompress now delegate to compressSync / decompressSync for cross-runtime compatibility
- ZipTools.createZip and ZipTools.extractZip now use zipSync/unzipSync and return Buffers
- Add TEntryFilter type to ts/interfaces.ts for fluent API entry filtering
- Minor readme.hints.md updates and small whitespace tidy in tests

2025-11-25 - 5.0.0 - BREAKING CHANGE(SmartArchive)

Refactor public API: rename factory/extraction methods, introduce typed interfaces and improved compression tools

- Renamed SmartArchive factory methods: fromArchiveUrl -> fromUrl, fromArchiveFile -> fromFile, fromArchiveStream -> fromStream; added fromBuffer helper.
- Renamed extraction APIs: exportToFs -> extractToDirectory and exportToStreamOfStreamFiles -> extractToStream; stream-based helpers updated accordingly.
- Export surface reorganized (ts/index.ts): core interfaces and errors are exported and new modules (bzip2tools, archiveanalyzer) are publicly available.

- Introduced strong TypeScript types (ts/interfaces.ts) and centralized error types (ts/errors.ts) including Bzip2Error and BZIP2_ERROR_CODES.
- Refactored format implementations and stream transforms: GzipTools/GzipCompressionTransform/GzipDecompressionTransform, ZipTools (ZipCompressionStream, ZipDecompressionTransform), TarTools improvements.
- BZIP2 implementation improvements: new bit iterator (IBitReader), clearer error handling and streaming unbzip2 transform.
- Updated tests to use the new APIs and method names.
- Breaking change: public API method names and some class/transform names have changed — this requires code updates for consumers.

2025-11-25 - 4.2.4 - fix(plugings)

Migrate filesystem usage to Node fs/fsPromises and upgrade smartfile to v13; add listFileTree helper and update tests

- Bumped dependency @push.rocks/smartfile to ^13.0.0 and removed unused dependency `through`
- Replaced usages of smartfile.fs and smartfile.fsStream with Node native fs and fs/promises (createReadStream/createWriteStream, mkdir({recursive:true}), stat, readFile)
- Added plugins.listFileTree helper (recursive directory lister) and used it in TarTools.packDirectory and tests
- Updated SmartArchive.exportToFs to use plugins.fs and plugins.fsPromises for directory creation and file writes
- Updated TarTools to use plugins.fs.createReadStream and plugins.fsPromises.stat when packing directories
- Converted/updated tests to a Node/Deno-friendly test file (test.node+deno.ts) and switched test helpers to use fsPromises
- Added readme.hints.md with migration notes for Smartfile v13 and architecture/dependency notes

2025-11-25 - 4.2.3 - fix(build)

Upgrade dev tooling: bump @git.zone/tsbuild, @git.zone/tsrun and @git.zone/tstest versions

- Bump @git.zone/tsbuild from ^2.6.6 to ^3.1.0
- Bump @git.zone/tsrun from ^1.3.3 to ^2.0.0
- Bump @git.zone/tstest from ^2.3.4 to ^3.1.3

2025-08-18 - 4.2.2 - fix(smartarchive)

Improve tar entry streaming handling and add in-memory gzip/tgz tests

- Fix tar entry handling: properly consume directory entries (resume stream) and wait for entry end before continuing to next header
- Wrap tar file entries with a PassThrough so extracted StreamFile instances can be consumed while the tar extractor continues
- Handle nested archives correctly by piping resultStream -> decompressionStream -> analyzer -> unpacker, avoiding premature end signals
- Add and expand tests in test/test.gzip.ts: verify package.json and TS/license files after extraction, add in-memory gzip extraction test, and add real tgz-in-memory download+extraction test
- Minor logging improvements for tar extraction flow

2025-08-18 - 4.2.1 - fix(gzip)

Improve gzip streaming decompression, archive analysis and unpacking; add gzip tests

- Add a streaming DecompressGunzipTransform using fflate.Gunzip with proper _flush handling to support chunked gzip input and avoid buffering issues.
- Refactor ArchiveAnalyzer: introduce IAnalyzedResult, getAnalyzedStream(), and getDecompressionStream() to better detect mime types and wire appropriate decompression streams (gzip, zip, bzip2, tar).
- Use SmartRequest response streams converted via stream.Readable.fromWeb for URL sources in SmartArchive.getArchiveStream() to improve remote archive handling.
- Improve nested archive unpacking and SmartArchive export pipeline: more robust tar/zip handling, consistent SmartDuplex usage and backpressure handling.
- Enhance exportToFs: ensure directories, improved logging for relative paths, and safer write-stream wiring.
- Add comprehensive gzip-focused tests (test/test.gzip.ts) covering file extraction, stream extraction, header filename handling, large files, and a real-world tgz-from-URL extraction scenario.

2025-08-18 - 4.2.0 -

feat(classes.smartarchive)

Support URL streams, recursive archive unpacking and filesystem export; improve ZIP/GZIP/BZIP2 robustness; CI and package metadata updates

- Add `exportToFs(targetDir, fileName?)` to write extracted `StreamFile` objects to the filesystem (ensures directories, logs relative paths, waits for write completion).
- Implement `exportToStreamOfStreamFiles` with recursive unpacking pipeline that handles `application/x-tar` (tar-stream Extract), `application/zip` (fflate Unzip), nested archives and `StreamIntake` for `StreamFile` results.
- Enhance `getArchiveStream()` to support URL/web streams (`SmartRequest`) and return Node Readable streams for remote archives.
- Make ZIP decompression more robust: accept `ArrayBuffer`-like chunks, coerce to `Buffer` before pushing to `fflate.Unzip`, and ensure `SmartDuplex` handling of results.
- Fixes and improvements to `bzip2/gzip/tar` tool implementations (various bug/formatting fixes, safer CRC and stream handling).
- Update CI workflows to use new registry image and adjust `npmci` install path; minor `.gitignore` additions.
- Package metadata tweaks: bugs URL and homepage updated, `packageManager/pnpm` fields adjusted.
- Documentation/readme expanded and polished with quick start, examples and API reference updates.
- Small test and plugin export cleanups (formatting and trailing commas removed/added).
- TypeScript/formatting fixes across many files (consistent casing, trailing commas, typings, `tsconfig` additions).

2025-08-18 - 4.1.0 -

feat(classes.smartarchive)

Support URL web streams, add recursive archive unpacking and filesystem export, and improve ZIP decompression robustness

- `ts/classes.smartarchive.ts`: add `exportToFs(targetDir, fileName?)` to write extracted `StreamFile` objects to the filesystem (ensures directories, logs relative paths, waits for write completion).
- `ts/classes.smartarchive.ts`: implement `exportToStreamOfStreamFiles` with recursive unpacking pipeline that handles `application/x-tar` (tar-stream Extract), `application/zip`

(fflate unzip), nested archives and StreamIntake for StreamFile results.

- `ts/classes.smartarchive.ts`: improve `getArchiveStream()` for URL sources by using `SmartRequest.create().url(...).get()` and converting the returned Web stream into a Node Readable stream.
- `ts/classes.ziptools.ts`: make ZIP decompression `writeFunction` more robust — accept non-Buffer chunks, coerce to Buffer before pushing to `fflate.Unzip`, and loosen the `writeFunction` typing to handle incoming ArrayBuffer-like data.

2024-10-13 - 4.0.39 - fix(core)

Fix dependencies and update documentation.

- Ensure package uses the latest dependencies
- Reviewed and grouped imports in TypeScript files
- Updated readme with advanced usage examples

2024-10-13 - 4.0.38 - fix(dependencies)

Update dependencies to latest versions

- Updated `@push.rocks/smartfile` to version 11.0.21
- Updated `@push.rocks/smartpromise` to version 4.0.4
- Updated `@push.rocks/smartstream` to version 3.0.46
- Updated `@push.rocks/smarturl` to version 3.1.0
- Updated `file-type` to version 19.5.0
- Updated `@git.zone/tsbuild` to version 2.1.84
- Updated `@git.zone/tsrun` to version 1.2.49
- Updated `@push.rocks/tapbundle` to version 5.3.0

2024-06-08 - 4.0.24 to 4.0.37 - Fixes and Updates

Core updates and bug fixes were implemented in versions 4.0.24 through 4.0.37.

- Repeated core updates and fixes applied consistently across multiple versions.

2024-06-06 - 4.0.22 to 4.0.23 - Descriptions and Fixes Updates

Efforts to update documentation and core features.

- "update description" in 4.0.22
- Updates to `tsconfig` and `npmextra.json` were performed.
- Ongoing core fixes.

2023-11-06 - 4.0.0 - Major Update with Breaking Changes

Introduction of significant updates and breaking changes.

- Transition to new version 4.0.0 with core updates.
- Break in compatibility due to major structural changes with core functionalities.

2023-07-11 - 3.0.6 - Organizational Changes

Structural reorganization and updates to the organization schema.

- Switch to new organizational schema implemented.

2022-04-04 - 3.0.0 - Build Updates and Breaking Changes

Major build update introducing breaking changes.

- Introduction of ESM structure with breaking changes.

2016-01-18 - 0.0.0 to 1.0.0 - Initial Development and Launch

Initial software development and establishment of core features.

- Project set-up including Travis CI integration.
- Launch of the first full version with code restructuring.
- Added callback support.

Revision #2

Created 2026-03-28 13:08:09 UTC by foss.global Team

Updated 2026-03-29 16:51:01 UTC by foss.global Team