

# TREEs™ File System

An embedded transactional file system with native flash support, providing the application with robust, fail-safe, data storage through a simple file interface. Features snapshot support and concurrent access management, all within a RAM footprint as low as 4 KiB.

## Features

**Transactional File System.** Write operations can be grouped into atomic write transactions so as to enforce strictly defined behaviours and guarantee data integrity, despite untimely interruptions such as power losses and errors.

**Snapshot support.** Snapshots provide a space-efficient way of saving the current file system state for later access, allowing for robust backup/recovery schemes.

**Concurrent read/write access management.** TSFS provides the application with read/write transactions, enabling consistent and strictly defined concurrent read/write behaviour.

**Low RAM usage.** The minimum amount of RAM required by TSFS is 4 KiB, independent of the size of the underlying media or the number/size of stored files.

**Native flash support.** TSFS natively supports bare flash storage technologies. Therefore it does not require an extra flash adaptation layer, reducing resource usage, complexity and costs.

**Dynamic and static wear-levelling.** Flash support includes both static and dynamic wear-levelling, maximizing the flash device lifetime, independent of the application's access patterns.

**High-speed random write access.** Assuming 4 KiB long, aligned, random write accesses, uniformly distributed across a 4 GiB file, the average net write throughput (i.e. the amount of user data per unit of time) is at least 80% of the maximum physical write throughput, on all supported storage technologies (assuming I/O bound performances).

**Low worst case write latency.** Write accesses triggered by internal bookkeeping tasks (such as flash wear-levelling) are uniformly spread across user requested write operations, such as to prevent unpredictable delays at the application level.

**Multiple media support.** TSFS is supported on a variety of media, including NOR flash, SD/MMC and nonvolatile RAM-like technologies such as MRAM and FRAM.

**Virtually unlimited files and file sizes.** TSFS uses 64-bit wide sizes, addresses and file IDs to support even the most extreme storage size requirements.

**POSIX-like file and directory interface.** Even though TSFS has its own consistent and orthogonal interface style, the file and directory interface will look familiar to most developers as it is largely inspired by the POSIX interface.

**Consistent and robust error handling.** Functions that can fail always return an error as the C functions return value. TSFS tries to return to the caller on an error condition to give the application developer maximum flexibility in handling fatal errors.

**ISO/IEC 9899:1999 compliant C code.** TSFS is distributed in ISO/IEC 9899:1999 compliant source code.

## Advantages

**Reduced application complexity.** TSFS's transactional nature makes the application development easier by providing a centralized and uniform way of dealing with unexpected failures, avoiding complex, error-prone, ad hoc application-level recovery schemes.

**Reduced risk and time to market.** Because it can be deployed with minimal resource usage, on top of a wide array of storage technologies, TSFS enables flexible and resilient development processes, immune to last-minute hardware design modifications.

**High-reliability.** TSFS transactional design guarantees performant, fail-safe operations ready to support mission-critical applications.

**High-scaleability.** TSFS is designed to support a wide range of applications, ranging from low-power designs based on modest processors with a few megabytes of sparsely accessed storage, to even the heaviest workloads, involving terabytes of data and strict performance requirements.

**Progressive learning curve.** TSFS basic features are mostly exposed through a conventional, POSIX-like, file API, allowing for a quick initial approach and deployment.

**Versatile and future proof.** Thanks to its native flash support and modular design, TSFS can be deployed on top of almost any storage technology, including emergent technologies such as MRAM and FRAM.

**Platform and OS agnostic.** TSFS is not tied to any OS or platform and can be deployed anywhere, as long as the minimum requirements are met.

## What's Included?

TSFS is delivered with everything needed to start development right away.

**Source Code.** TSFS is delivered in source form.

**Media drivers.** TSFS comes complete with all the needed media drivers.

**RTOS and platform integration.** TSFS comes ready to run with the provided RTOS and platform integration.

**Getting Started Guide.** The getting started guide includes instructions on installing and running the delivered source and projects as well as a summary of the delivered components and documents.

**Development Project.** Each release package contains a development project for the IDE and toolchain selected by the developer. The development project also includes the debug configuration.

**User & Reference Manuals.** The user manual as well as the API reference manual for TSFS are included.

## Supported Platforms

Through its integration with the BASEplatform, TSFS is designed to support the widest range of platforms, toolchains and RTOSes without compromising features and performance. Platform support can range from small low power MCU to large heterogeneous multi-core System on Chip.

### General Requirements

**CPU.** 32 or 64-bit architecture is recommended for optimal performance.

**RAM and ROM.** Requirements are dependent on the underlying media technology and performance expectations. The strict minimum is 4 KiB but more RAM can be used to improve overall performances. ROM usage is approximately 30 KiB.

**Supported media.** RAM disk, NOR, SD/MMC, M/F-RAM.

**Toolchain.** ISO/IEC 9899:1999 (a.k.a C99) compliant compiler.

## About JBLoopen

Founded in Montreal, Canada in 2016, JBLoopen provides embedded software components and consulting services. JBLoopen is specialized in offering platform support products and services as well as RTOS and third party library integration.