

Main

Welcome to the TSAN Project!

The goal of this [Project](#) is to explore and incubate a Thread SANitizer (TSAN) feature that could be integrated into the HotSpot JVM and the JVM Tool Interface (JVM TI). This includes working, evaluating, and incubating a [Thread Sanitizer implementation](#) for Java.

The rest of this page will provide basic information for downloading, building, and testing Java TSAN.

Prerequisites

Java TSAN has only been tested on 64-bit Linux. It may work on other platforms that [LLVM TSAN](#) supports with some build changes.

Java TSAN is dependent on the [C/C++ ThreadSanitizer implementation for GCC/LLVM](#). Java TSAN requires [a new API](#) for proper symbolization of Java stack traces, which is only available in update-to-date versions of C/C++ ThreadSanitizer. This requires building OpenJDK with LLVM 7.0 or later, or GCC 9 or later. Currently we recommend developers and testers use [LLVM 8.0](#).

Building

Java TSAN implementation is in the "tsan" branch in the Git repository. Developers need to clone this branch:

```
git clone --branch tsan https://github.com/openjdk/tsan.git jdk-tsan
```

Newer toolchains come with new warnings and the codebase hasn't been cleaned up for those warnings yet, so `--disable-warnings-as-errors` is needed.

Java TSAN exists as a JVM feature "tsan", which is enabled by default on the "tsan" branch in the repository. It can be disabled with the configure flag `--with-jvm-features=-tsan`.

To build OpenJDK with LLVM, pass the configure flags `--with-toolchain-type=clang --with-toolchain-path=<path_to_llvm_root_dir>/bin`.

Running

Note: the method of including the TSAN runtime in the JVM is not finalized and could change in the future.

Run Java with TSAN enabled:

```
./images/jdk/bin/java -XX:+ThreadSanitizer
```

Testing

The tests for TSAN are at `test/hotspot/jtreg/tsan/`

```
make test TEST="hotspot/jtreg/tsan"
```

Known Issues

- We have only tested TSAN with the G1 garbage collector. TSAN may not work correctly with other garbage collectors.
- For a moderately scaled application, TSAN will likely report data races from OpenJDK's own libraries. Many of these races are benign, but these race reports hurt TSAN's usability. You could specific a file via `-XX:ThreadSanitizerIgnoreFile` to suppress such races (refer to [this doc](#) for its format). We plan to publish a list of commonly suppressed races in the future.

Resources

- [TSAN Project](#)
- [TSAN JEP proposal](#)
- [Repositories](#)
- Mailing list: [tsan-dev](#)

Recent space activity



[Man Cao](#)

Space contributors

- [Man Cao](#) (201 days ago)
- [Arthur Eubanks](#) (2022 days ago)
- [Jean Christophe Beyler](#) (2119 days ago)
- [Iris Clark](#) (2159 days ago)

[Main](#) updated Jul 24, 2024 [view change](#)



[Arthur Eubanks](#)

[Main](#) updated Jul 29, 2019 [view change](#)



[Jean Christophe Beyler](#)

[Main](#) updated Apr 23, 2019 [view change](#)



[Iris Clark](#)

[Main](#) updated Mar 14, 2019 [view change](#)