

# LiveRecorder for Java Data Sheet

undo™

## Record. Replay. Resolve.

LiveRecorder for Java is a software failure replay system that provides application developers with a powerful Record and Replay toolkit.

By eliminating the usual guesswork involved in software application failure diagnosis, LiveRecorder significantly accelerates root cause detection and overall Mean Time to Resolution (MTTR) compared with slower traditional methods of debugging.

50%

of software engineers' time is still spent debugging

91%

of software developers admit to having unresolved defects because they can't reproduce them.\* **Reproducibility** is the fundamental problem in software defect resolution

PROBLEM	SOLUTION
Increasing software application architecture/ orchestration <b>complexity</b> makes observability a challenge.	LiveRecorder provides better observability and <b>visibility into what a failing process actually did.</b>
<b>Test failures</b> can result in deployments being full of ticking time bombs.	<b>LiveRecorder records a program's execution down to instruction level.</b> A recording supplies all the context needed for developers to determine root cause quickly.
<b>Hard to reproduce, intermittent,</b> software failures. Recreating the conditions under which the software failed can be near impossible.	A LiveRecorder recording <b>eliminates the need for reproducibility</b> by providing a 100% reproducible test case of a software failure ready for analysis & debugging.
<b>Traditional caveman debugging techniques</b> (printf, logging etc) rely on a lot of guesswork and are needlessly <b>time-consuming.</b>	LiveRecorder includes a powerful integrated reversible debugger. Play recordings forwards and backwards to analyze internal program state.

## Key Features

### RECORD

Record programming errors (capture all non-deterministic data down to instruction level). Debug the recording file.

### REPLAY

Reversible debugger - play recordings forwards and backwards to analyze internal state. Breakpoints, bookmarks, notifications make it easy.

### SUPPORTED LANGUAGES

Java, Kotlin, Scala on Linux x86\_64. Compatible with all mainstream Linux distributions, containerised environments and hypervisors.

### INTEGRATIONS

Seamless integration into your Java application & development workflow via Jenkins, Maven and Gradle integration and IntelliJ IDE support (VSCode & Eclipse coming soon).

### CI/CD

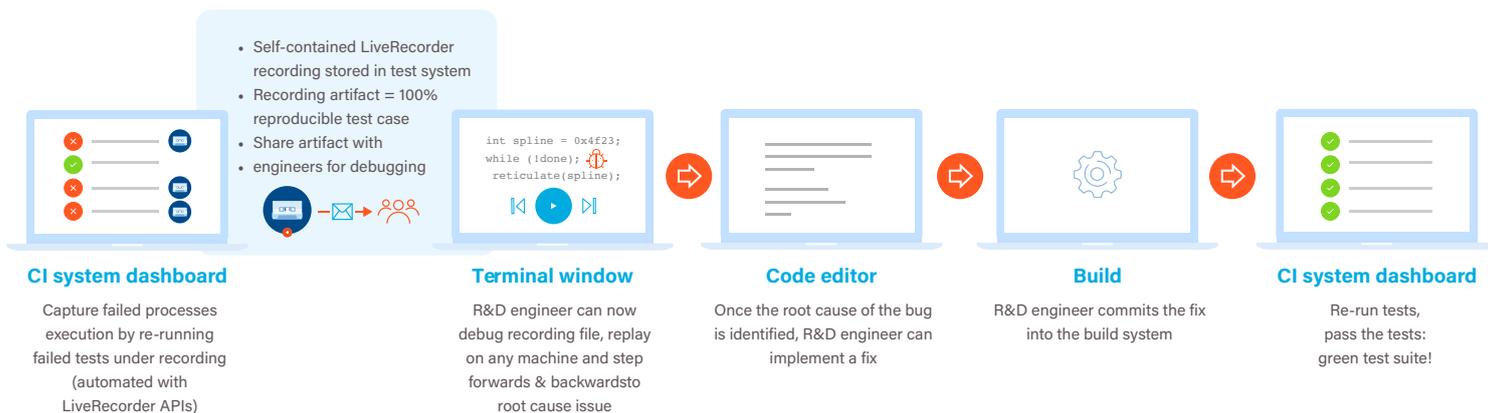
Configure your CI pipeline to only record failing tests (not all tests) and generate 100% reproducible test cases of the failures. Debug the recordings.

### FASTER THAN YOU THINK

Expect a 2x to 5x slowdown, depending on workload.

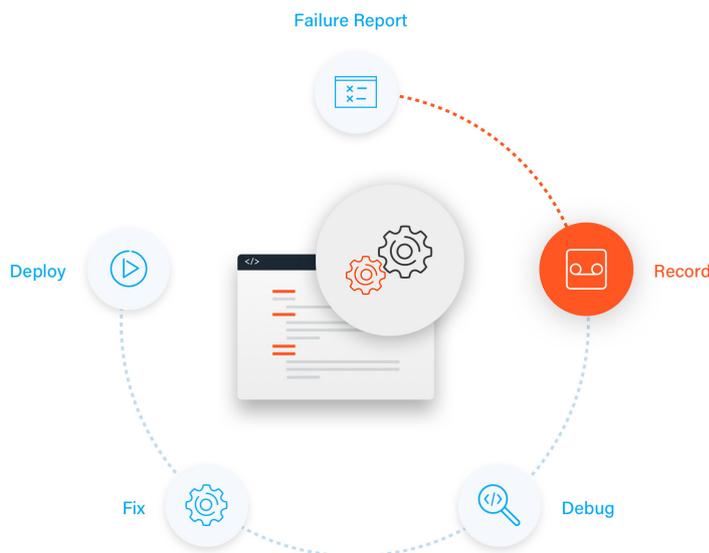
# LiveRecorder for development

Software development testing and quality assurance is "shifting left" to earlier stages of development. LiveRecorder can be deployed to accelerate bug defect resolution across all phases of the software development lifecycle. Here's how, for example, LiveRecorder can fit into your CI workflow.



# LiveRecorder for production

If software fails in production, reducing MTTR is essential to minimizing customer disruption. Here's how LiveRecorder enables developers to simply record the failing software (no reproducibility needed) and to analyze and debug the recording without disrupting the live environment.



## Trusted by Industry Leaders



## Undo: Making Software Reliable

Undo is the leading software failure replay platform provider for engineering teams building complex systems. Our core platform, LiveRecorder, is used to reproduce and fix software bugs faster, accelerate software delivery, and reduce engineering costs. With offices in Cambridge,

UK and San Francisco, CA, Undo's platforms are used by thousands of software engineers across leading technology companies including SAP, Juniper, Cadence Design Systems, Micro Focus, and Mentor (a Siemens business). For more information, visit us online at [undo.io](https://undo.io)