



Eliminating IT Risk in Multi-threaded Applications

Executive Summary

ThreadSafe, the leading Java concurrency analysis product, was deployed within a global top-10 investment bank in an application that is both high profile and business critical. The application is highly multi-threaded and therefore an insight into the runtime behaviours of the code is vital in guaranteeing the required high level of stability of the application. *ThreadSafe* identified a number of code vulnerabilities which had the potential to threaten the operational stability of the customer service. Following the initial code analysis and remediation project, the client has integrated *ThreadSafe* into its quality management approach to ensure the mitigation of risks as the application continues to be developed.

The Project

The application analysed by Contemplate's *ThreadSafe* is an Algorithmic Trading Execution system, processing customer orders and executing trades across a variety of assets including Equities, Futures, Options and FX. To deliver the required level of business performance the application has to be highly concurrent.

The application uses a custom framework to provide concurrency, with extensive use of thread confinement to maintain thread-safety. The project is built using Ant.

The Problem

The application has a high degree of change to react to new trading business requirements in a highly competitive area of the bank's operations. Release cycles are planned bi-weekly but are often made more frequently than that. With a global team of 15 development staff, coordination of changes and assurance of enforcement of architectural integrity is challenging in such a multi-threaded application.

Consequences of system error are well-managed, but occurrence of issues can have serious operational consequences in terms of roll-back and reprocessing, and there remains a residual risk that problems which remain undetected could lead to reputational damage and franchise risk. Serious issues have the potential to become regulatory matters. Because of this, the application team look to maintain the highest level of code quality, and *ThreadSafe* provided the ability to quickly identify the most insidious and difficult to spot vulnerabilities, both within the framework and within the core business application logic. By running *ThreadSafe*, the client was able to identify issues that were impossible to find using other techniques such as peer-review, traditional system and UAT testing, or using the first generation Static Analysis tools currently deployed in the bank.

Initial results identified potentially important race conditions and related problems in the core code and in the framework which were immediately fed back into the development process at the investment bank for repair.



ThreadSafe

ThreadSafe identifies concurrency problems in Java code, helping developers to avoid costly software bugs and helping to reduce technical debt and operational risk. It is built on Contemplate's Core Analysis Toolkit (CAT) and is designed to integrate analysis seamlessly with developer workflow, including desktop tools such as Eclipse and build server environments.

Client Comment

The Director responsible for the global team commented: *"Code quality is hugely important to us. With the deep analysis of run-time behaviours in our code that ThreadSafe provides, we are able to offer a safer and more secure service to our external and internal clients. ThreadSafe has helped us to easily remove some of the serious risks in our key trading application, and we plan to integrate ThreadSafe into our quality management approach for all future system releases and upgrades."*