TeamCity Features (2024.03 – 2025.03) – Detailed Overview

## User Interface (UI) Updates

### Settings Toggle

The old Edit/View buttons are replaced with a persistent toggle, and colors now help users visually distinguish between editing and viewing mode.

A screenshot of a computer

AI-generated content may be incorrect.

### Refined Layout

Header sections for builds, projects, and configurations now include compact and useful summary information (like branch name and build time) with a more modern UI.

A screenshot of a computer

AI-generated content may be incorrect.

### Problems Tab Redesign

The Problems page now consolidates failed tests, build issues, and error grouping with better filters and visual design—making issue tracking and debugging easier.

A screenshot of a computer

AI-generated content may be incorrect.

## Build Monitoring & Issue Tracking

### Problems Page

The new unified view helps track and resolve build and test issues across all configurations with filters, grouping, and visibility improvements make it a vital debugging tool.

A screenshot of a computer

AI-generated content may be incorrect.

## Version Control System (VCS) Improvements

### Non-Recursive Submodule Checkout

A new option enables one-level submodule checkout. This avoids deep recursive fetching of nested repositories, saving time and bandwidth.



## CI/CD Workflow Enhancements

### Pull Request Filters

TeamCity now supports +pr: and -pr: filters to include or exclude pull requests based on source branch, target branch, or author.

A screenshot of a computer

AI-generated content may be incorrect.

## Artifact & Dependency Management

### Optional Artifact Dependencies

Builds can define optional artifact dependencies using **?:** in the rule. If the dependent artifact is missing, the build still proceeds, making workflows more resilient.

## Build & Test Enhancements

### .NET Test Retry

TeamCity now supports retrying failed .NET tests during the same build. This helps identify flaky tests and improves confidence in test results without needing a separate re-run build.

A screenshot of a computer

AI-generated content may be incorrect.

### New dotCover Runner

A dedicated dotCover runner enables automatic merging of multiple .NET test coverage snapshots into a single report, simplifying coverage tracking in complex pipelines.

A screenshot of a computer

AI-generated content may be incorrect.

### Gradle Configuration Cache

Gradle builds now benefit from configuration caching, significantly reducing build time by skipping re-evaluation of unchanged build scripts. Stores information locally on the build agent’s disk.

### Bootstrap Steps

New “bootstrap” steps can be configured to run before source checkout. These are useful for setting up authentication or environment conditions required before the main pipeline logic.

A screenshot of a computer

AI-generated content may be incorrect.

### NUnit/NAnt Runners Deprecated

The legacy NUnit and NAnt runners are deprecated and will be moved to plugins.

## Security & Secrets Management

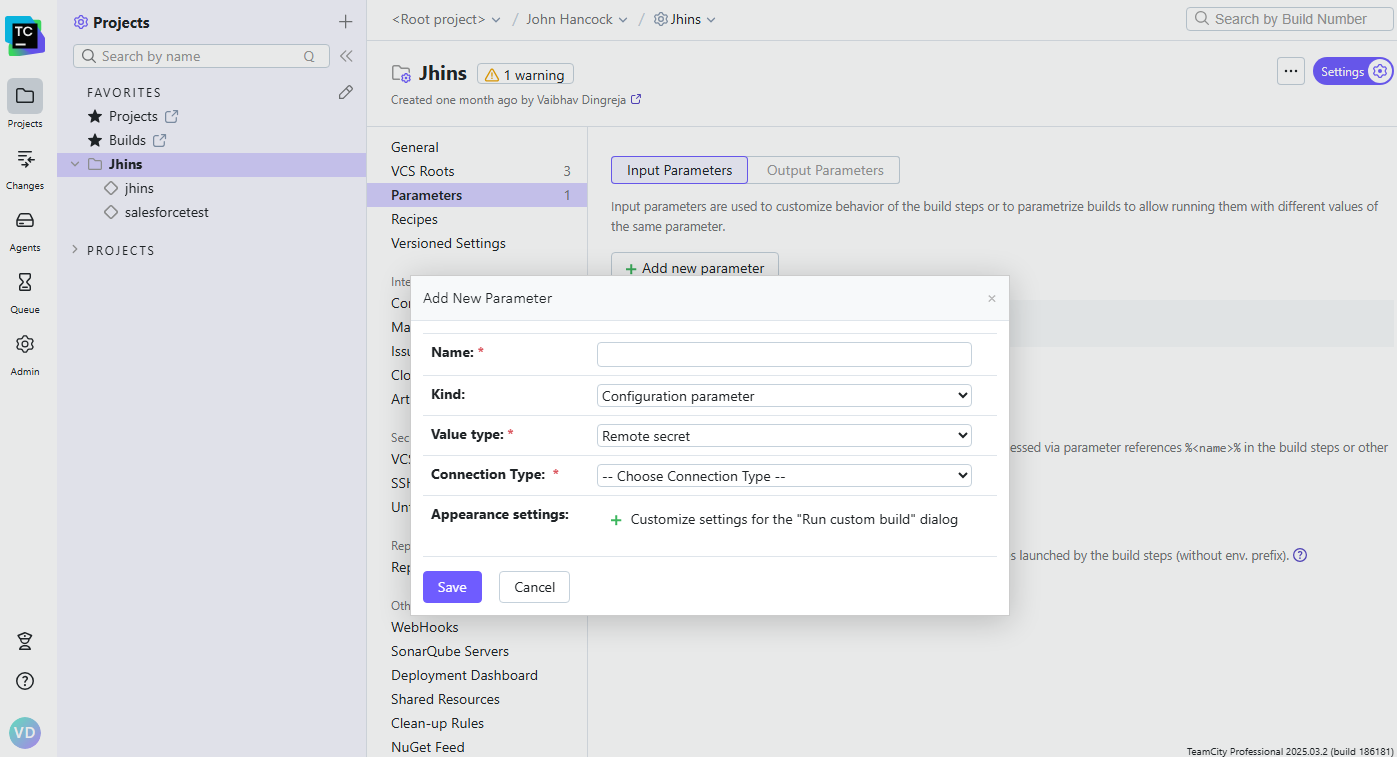
### Bundled Hashi Corp Vault Plugin

The Hashi Corp Vault plugin is now bundled with TeamCity out of the box, removing the need for manual installation. It allows secure retrieval of secrets (tokens, passwords, etc.) at build time.

A screenshot of a computer

AI-generated content may be incorrect.

### Remote Secret Parameters

You can now define parameters that fetch values securely from remote secret providers like Hashi Corp Vault. These are evaluated at runtime, avoiding hardcoding secrets in the build configuration

### OAuth Security Enhancement

A new setting enables a unique redirect URL for each OAuth connection, which mitigates the risk of mix-up attacks when multiple connections use the same provider

A screenshot of a computer

AI-generated content may be incorrect.

## Containerization & Docker Integration

### Run in Docker Feature

A new “Run in Docker” option at the configuration level allows all steps in a build to run in a single container, simplifying dependency and environment management.

A screenshot of a computer

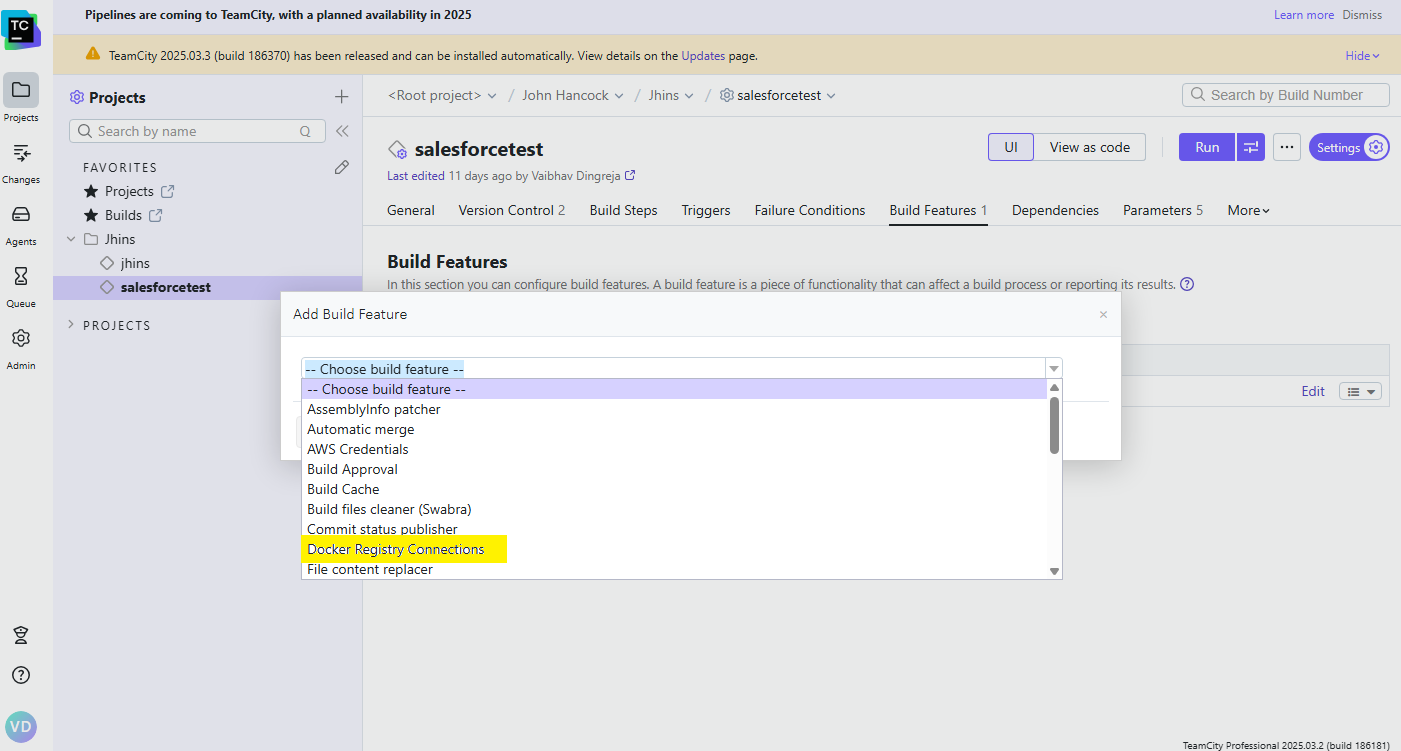
AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

### Docker Registry Connections

Formerly called “Docker Support,” this feature has been renamed and refocused on managing container registry credentials and cleanup more explicitly.



## Build Configuration & Parameter Management

### New Parameter Dialog

The Add/Edit Parameter UI is redesigned with a cleaner layout, better validation, and easier access to advanced options like secret parameters.

A screenshot of a computer

AI-generated content may be incorrect.

### Input and Output Parameters

Build parameters are now clearly labeled as input or output. Output parameters can be shared with dependent builds using dep.<config>.<param>, and sharing is explicitly defined.

A screenshot of a computer

AI-generated content may be incorrect.

### Custom Location for Versioned Settings

When using Kotlin DSL or XML in VCS, you can now specify a custom directory path for versioned settings—especially helpful in mono repo setups.

A screenshot of a computer

AI-generated content may be incorrect.