



# INTELLIGENCE USABILITY PRODUCTIVITY

## YOUR SOLUTION FOR CONTINUOUS INTEGRATION AND EFFECTIVE BUILD MANAGEMENT

Web-based Administration and Build Process Monitoring

Code Quality Features

Eclipse, MS Visual Studio, and IntelliJ IDEA Integration

.NET and Java Build Runners

Support of industry-wide Version Control Systems

Open API and Extensibility



**Develop with pleasure!**

## TeamCity – Your Solution for Continuous Integration and Effective Build Management

TeamCity is a distributed build management and continuous integration system that allows your team to run and monitor your software building process while improving your team communication and the integration of changes. With TeamCity, you'll always know when a build breaks, and exactly which code broke it, so you won't need to waste time hunting down the causes of failed builds, missed deadlines, low quality software, and poor morale.

TeamCity works the way you want it to, and you don't need to change your usual development style: code in your favourite IDE, commit changes to your VCS, and get feedback on the results of integrating your changes using TeamCity's web interface or your preferred means of notification.

The screenshot shows the TeamCity web interface. At the top, there are tabs for 'Projects', 'My Changes', 'Agents (22)', and 'Build Queue (12)'. The 'Build Queue' tab is active. Below it, the 'Projects' section is visible, showing a list of projects. One project, 'Agra', is expanded to show a build named 'IntegrationBuild' which is currently running. The build status is 'Running 1 build' and it has a 'Run' button. Below this, another build 'IntegrationBuild (MySQL)' is shown with a status of 'Running 1 build' and a 'Run' button. The build details for 'IntegrationBuild (MySQL)' show 'Tests failed: 1' and a duration of '09 Mar 15:23 - ...'. The 'IntegrationBuild' build shows 'Checking for changes' and a duration of '09 Mar 15:56 - ...'.

## TeamCity's Technologies

To ensure the best user experience, we employ cutting-edge web technologies such as Ajax and Web 2.0 with Java, to make TeamCity a cross-platform and IDE-independent solution.

### Build Agents and "Build Grid"

TeamCity uses distributed "Build Agents" to manage the building of your projects. You can customize the number of computers you use as Build Agents, and control them builds, directly from TeamCity's web interface. The process of installing and customizing the Build Agents is easy and straightforward, and their update is completely automated.

TeamCity can leverage the available computing resources of an entire organization with its Build Grid concept, which employs currently-unused resources from multiple computers, any of which can run multiple builds and/or tests at a time, for single or multiple projects across your company.

The screenshot shows the 'Agents' section of the TeamCity web interface. It displays a list of agents with their status and running builds. The agents are: 'Agent mps-unit' (Enabled, Running build: Selena :: Main, Tests passed: 6268), 'resharper-tests' (Enabled, Running build: Selena :: Main, clear.classes), 'dotTraceTests\_amd64' (Enabled, Running build: dotTrace-Main :: PerformanceTests, Tests passed: 2936), and 'dotTraceTests\_ia32' (Enabled, Running build: dotTrace-Main :: StressTests\_ia32, Tests failed: 1(1 new)).

### Build Queue

TeamCity processes builds in its "Build Queue" – a list of pending builds which you can manually reorder by dragging-and-dropping.

The screenshot shows the 'Build Queue' section of the TeamCity web interface. It displays a list of builds with columns for 'Build configuration name', 'Added to queue', 'Triggered by', and 'Can run on'. The builds are: 'Omea :: Omea Pro' (Added to queue: 06 Mar 20:33, Triggered by: perforce, Can run on: 1 agent), 'Selena :: Duplicates' (Added to queue: 08 Mar 22:32, Triggered by: Scheduler, Can run on: 1 agent), and 'dotTrace-Main :: PerformanceTests' (Added to queue: 09 Mar 15:46, Triggered by: dotTrace-Main, Can run on: 1 agent).

### Web-based User Interface

A rich web interface enables the efficient administration and automation of common user tasks:

- get quick feedback on the code changes you introduce
- access the build artifacts in just one click
- view the source code changes pending for the next build
- configure your own view of the projects dashboard
- view the files differences on the web just like in the IDE and navigate to the project source base from the web

The screenshot shows a diff view of a log file in the TeamCity web interface. The diff compares two versions of the log file, 'Before' and 'After'. The 'Before' version shows a sequence of commands and their outputs, including 'nsExec::ExecToLog', 'Pop \$0', 'StrCmp \$0 "0" InstallOk', 'MessageBox MB\_ABORTRE', 'Failed to install', '/SD IDIGNORE IDIGNO', 'Quit', 'InstallOk:', 'ClearErrors', 'Call SaveUserSettingsDi', 'WriteRegStr HKLM "SOFTW', 'WriteRegStr HKLM "SOFTW', 'nsExec::ExecToLog', and 'nsExec::ExecToLog'. The 'After' version shows the same sequence of commands and outputs, but with some changes in the output, including 'nsExec::ExecToLog', 'Pop \$0', 'StrCmp \$0 "0" InstallOk', 'MessageBox MB\_ABORTRE', 'Failed to install', '/SD IDIGNORE IDIGNO', 'Quit', 'InstallOk:', 'ClearErrors', 'Call SaveUserSettingsD', 'WriteRegStr HKLM "SOFT', 'WriteRegStr HKLM "SOFT', 'nsExec::ExecToLog', and 'nsExec::ExecToLog'.

### Supported Web Browsers

Use your favorite browser when working with TeamCity: MSIE 6.0 and 7.0, Mozilla Firefox 1.5.x and 2.0, Opera 8.5.x and 9.0, and Safari 2.0.x.

### Build Runners

Build your projects from the command line or use your favorite build tools, such as:

- Ant
- MSBuild
- Maven
- Microsoft Visual Studio 2005 and 2003 solutions
- NAnt
- (for Visual Basic and C#)

Plus, analyze your code quality remotely with Duplicator and Inspections build runners for IntelliJ IDEA projects.

### Supported VCSs

TeamCity supports such industry-wide version control systems as:

- Perforce
- CVS
- ClearCase
- Team Foundation Server
- Subversion
- Visual SourceSafe 2005 and 6.0

### Testing Frameworks

TeamCity integrates with JUnit and TestNG testing frameworks for Java, and NUnit for .NET platform solutions to enable the most effective and thorough testing of your software.

## Software Build Management Features

### The Server Possibilities

TeamCity provides multiple means for setting up an effective build process workflow:

- projects cloning
- linking issues in bug tracking system with comments for VCS commits
- dependent builds triggering
- LDAP integration
- checkout of the project sources on Build Agents

### Monitoring the Building Process

Continuous integration and testing processes just got a lot easier. Monitor your projects and changes integration status in TeamCity's web interface, or directly through your IDE to stay on top of things.

### Handling Failed Builds

When multiple developers submit multiple changes, and builds start failing, everyone thinks that someone else is dealing with the problem, so nothing gets done. That's not a problem anymore: TeamCity first identifies the build that failed, and then someone can take responsibility.

### Taking responsibility

When your changes cause the build to fail, you are notified almost immediately. You then have the option to take responsibility for it, so your teammates know that someone is making the necessary fixes.

|                  |  |
|------------------|--|
| Last build #5619 | Result: <span style="color: red;">Failure</span> <span style="color: blue;">Changes (1)</span> |
| Responsible:     | <i>You are responsible</i> <a href="#">Done!</a> <a href="#">Give up</a>                       |
| Time:            | 21 Nov 13:53 - 14:06 (12m:42s)   |
| Agent:           | unit-157   |
|                  | Comment: I am going to fix it in the next build.   |

### "First failed in"...

When a new test fails, TeamCity immediately shows it on the *Build Results* page. You can view the detailed description of problems which occurred or navigate to the test code in the IDE right from the web browser.

1 tests failed (1 new)

**NAntRunnerSendMessageTest.testProgress** [Open in IDE](#)

```
junit.framework.ComparisonFailure:  
<myStatus>1</myStatus>  
<myTimestamp>CENSORED</myTimestamp>
```

[Show details >>](#)

First failed in this build with 1 change

### "Already fixed in"...

As soon as the developer fixes the failed test, it is reflected in TeamCity's user interface: the test name is crossed out and there is a check mark near it.

1 tests failed (1 new)

~~NAntRunnerSendMessageTest.testProgress~~ [Open in IDE](#)

```
junit.framework.ComparisonFailure: expected:  
<myStatus>1</myStatus>  
<myTimestamp>CENSORED</myTimestamp>
```

[Show details >>](#)

Already fixed in build #3341

First failed in this build

## Navigating to Stacktraces, Duplicates, and Accessing Your Project Source Base from the Web

Navigate from TeamCity's web interface to the server-side code analysis results, duplicates and file differences, and the code of failed unit tests in your IDE.

## Team Communication

TeamCity allows to make your team a self-organizing one. Responsibilities are no longer handed to separate team members from the outside. If nobody fixes the failed builds, this fact is evident to all team members. But as soon as someone takes responsibility for breaking the build, the entire team is automatically informed that someone is working on the fix, and when the fix is complete.



Developers are always aware on what's going on in the project code base using TeamCity's web interface, or receiving notifications via Jabber/XMPP protocol, in their IDE, or in Windows System Tray.

## Using Code Quality Features

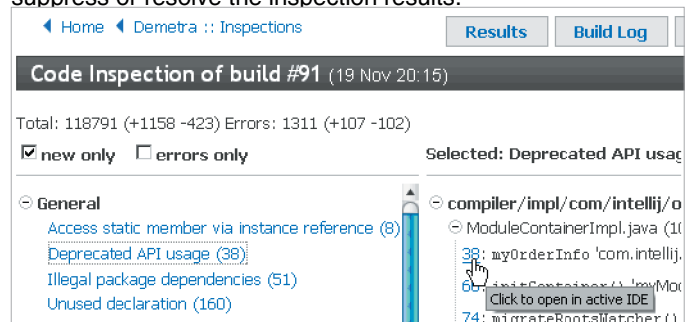
With TeamCity's code quality features you can perform such time-consuming operations like code analysis, review the code coverage and catch code duplicates on the web, thus, your own computer never slows down.

### Remote Code Analysis

TeamCity comes complete with a set of more than 600 IntelliJ IDEA inspections which you can use to remotely analyze:

- unused and unreachable code,
- declaration redundancies,
- performance issues,
- dead code.

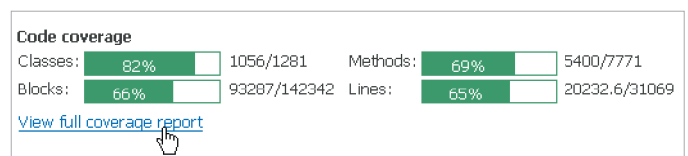
Check out the results of your remote code analysis in your browser, and then zoom right into the IDE where you can suppress or resolve the inspection results.



### Reviewing the Code Coverage

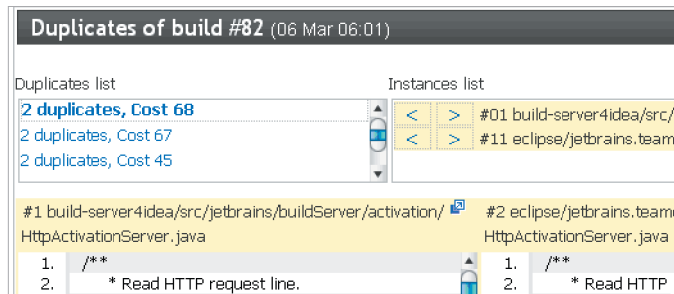
Review your code coverage statistics both on the web and within IntelliJ IDEA, so you'll always know the exact coverage of your project's code base with unit tests.

Code coverage is available for Ant build runner and IntelliJ IDEA projects. TeamCity's code coverage analysis is based on the EMMA open-source toolkit.



## Catching Code Duplicates

We have “transferred” IntelliJ IDEA’s powerful mechanism of searching for code duplicates on the server. TeamCity searches for the code duplicates remotely on one of the available build agents and provides an easy-to-navigate web report on found repetitive blocks of your code and enables one-click access to the IDE where you can analyze the discovered issues.



## Integration with IntelliJ IDEA, Eclipse, and MS Visual Studio

TeamCity integrates with the most popular IDE’s via plugins making your IDE not merely an IDE but a “remote control” for many TeamCity’s features:

- build status notifications
- one-click navigation to the web interface
- taking responsibility
- starting and stopping builds
- performing remote run and pre-tested (delayed) commit
- viewing tests results

### Remote Run for Personal Builds

The Remote Run command helps you to integrate and test your modified code. First, your modified files are submitted to the server, bypassing the version control system. Then, TeamCity obtains the latest source files from the project code base and creates a “personal” build to test how successful your changes are just the same way as a “regular” build. This allows you to quickly find out whether your changes, if submitted, could break the build.

### Pre-tested (Delayed) Commit

The pre-tested or delayed commit is an extension of the Remote Run feature. Submitted code changes first go through testing. If all tests succeed, TeamCity can automatically commit the changes to version control, or you can confirm the commit, and the code base changes will be integrated into the next project build. If any test fails, the code is not committed, the process of the entire team is not affected, and the developer can safely introduce the necessary fixes.

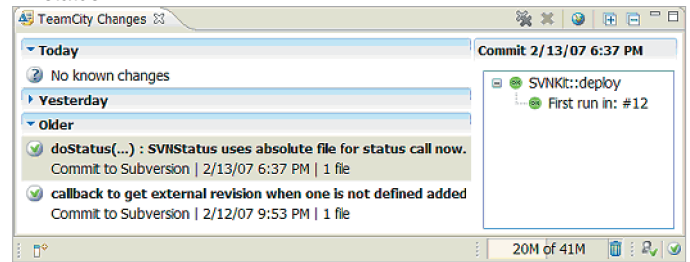
### IntelliJ IDEA

With TeamCity integration plugin for IntelliJ IDEA you can examine the results of code inspections and code duplicates, view the code coverage, get notifications about the build process and much much more...

## Eclipse

TeamCity integration plugin uses Subclipse, the SVN integration for Eclipse which allows:

- running both “personal” and “regular” builds and viewing the changes status

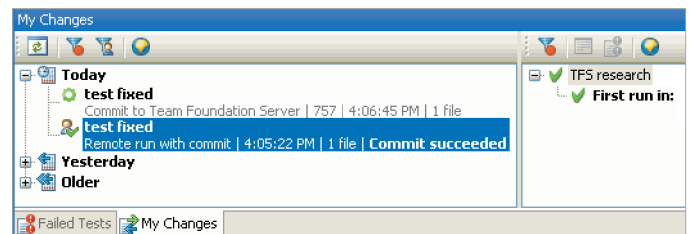


- navigating from the build logs opened within Eclipse to files referenced in build log and from TeamCity web client to Eclipse
- taking responsibility right in the IDE
- exploring the changes introduced in the source code and comparing the local version with the latest version in the project repository
- configuring watched projects and notification rules

## Microsoft Visual Studio 2005

TeamCity add-in integrates with Microsoft Team Foundation Server. After installing the plugin you can:

- perform both a remote run and delayed commit to the project code base managed by Team Foundation Server
- take responsibility for the failed build
- view the build logs
- navigate from the results of failed tests in TeamCity’s web-interface to Visual Studio editor



## Open API and Extensibility

TeamCity’s API allows you to extend it in different areas:

- User’s authentication mechanisms
- Version Control Systems
- Build Runners
- Notification services
- Web interface
- Build triggering options

**Improve the development and collaboration  
in your team and develop with pleasure using  
TeamCity!**

<http://www.jetbrains.com/teamcity/index.html>

**Headquarters and International Sales:**  
sales@jetbrains.com

JetBrains s.r.o. Klanova 9/506 14700  
Prague Czech Republic  
Tel: +420 2 4172 2501  
Fax: +420 2 6171 1724

**North American Sales:** sales.us@jetbrains.com

**East Coast**  
33 South Main St. Suite 2-A  
Medford, NJ 08055  
Tel: +1 609 714 7883  
Fax: +1 609 714 7886

**West Coast**  
1670 So. Amphlett Blvd. Suite 214  
San Mateo, CA 94402  
Tel: +1 650 378 8571  
Fax: +1 650 378 8591

**Canada**  
207 Barnes Street Kempville  
ON K0G 1J0  
Tel: +1 613 258 0575  
Fax: +1 613 258 5979