Release Notes for Agra, build 3776

> ⚠ Some features in this build may require manual reconfiguration of your builds.
> Please read carefully.

Incompatible changes

Database configuration

If your database configuration differs from default HSQLDB settings, you'll have to configure database in a separate database.properties file. This file should be located in the TeamCity data config directory (defaults $HOME\.BuildServer\config). After the first run of TeamCity template files database.mysql.properties and database.hsqldb.properties are be placed to this directory for reference. Copy one of them to database.properties and edit settings according to your database configuration.

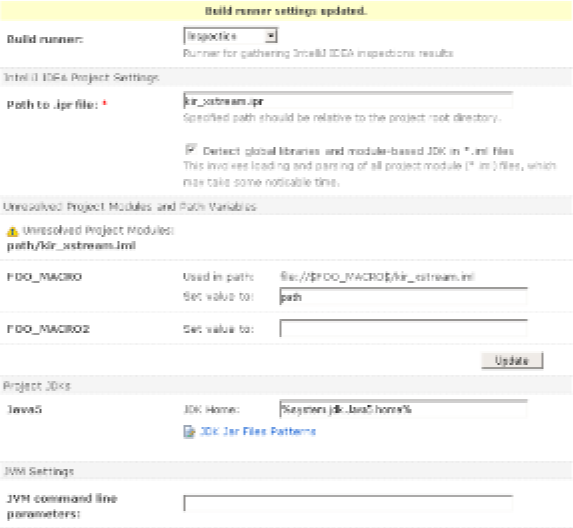
Mysql configuration example

```
driverName=com.mysql.jdbc.Driver
connectionUrl=jdbc:mysql://localhost/buildserver
connectionProperties.user=username
connectionProperties.password=pass

maxConnections=50
poolPreparedStatements=true
shutdownStatement=
shutdownCompactStatement=

connectionProperties.useLocalSessionState=true
connectionProperties.elideSetAutoCommits=true
connectionProperties.alwaysSendSetIsolation=false
```

Ipr, Inspections, Duplicates runners



Build runner settings updated.

Build runner: Inspection

Runner for gathering IntelliJ IDEA inspections results

IntelliJ IDEA Project Settings

Path to .ipr file: * kir_xstream.ipr

Specified path should be relative to the project root directory.

☑ Detect global libraries and module-based JDK in *.iml files

This involves loading and parsing of all project module (*.iml) files, which may take some noticable time.

Unresolved Project Modules and Path Variables

⚠ Unresolved Project Modules:
path/kir_xstream.iml

FOO_MACRO — Used in path: file://FOO_MACRO/kir_xstream.iml — Set value to: path

FOO_MACRO2 — Set value to:

Project JDKs

Java5 — JDK Home: %system.jdk.Java5.home%

JVM Settings

JVM command line parameters:

TeamCity provides enhanced dialog to setup builds which depend on IntelliJ IDEA's IPR project file. Unfortunately, it was impossible to convert old configuration to the new one, so after upgrade to this version of TeamCity you'll have to tweak your configuration manually.

To do it, visit build configuration page/build runner section. The first visit to this page may be slow, because TeamCity will download and analyze all project module (*.iml) files to detect used global libraries, path variables, jdk and so on. Subsequent visits will be much faster.

This page will provide options to configure project JDK, path variables, global libraries. If some module files could not be obtained from VCS, this will be shown as well.

Java Service Wrapper is used to run build agents as Windows service

There is no more agentd.exe in the new agent installation.

To run service user can use <agent>/bin/service.start.bat
To stop service user can user <agent>/bin/service.stop.bat
To install service <agent>/bin/service.install.bat
To uninstall service <agent>/bin/service.uninstall.bat

Installed agent will still use agentd.exe. Auto updates should work correctly as well. But, to migrate to the new service wrapper user should reinstall agent explicitly.

Using new service wrapper allows user to change any agent JVM parameters.

To upgrade agent from agentd service to java service wrapper:

- Uninstall agent
  - use uninstaller/add remove programs
  - ensure agent service is stoped
- Install new agent using any distribution you like (zip, java web start, windows installer

Windows Only:

- Make sure agent service is working under right user account. Using SYSTEM tend may affect MSBuild/Sln2005 configurations to fail.

Agent process will have a name TeamCityAgentService-windows-x86-32.exe

To work with net.exe utility please use the name TCBuildAgent. For
example:
net start TCBuildAgent - start agent

Links to build artifacts has changed

Now we use new format for links to build artifacts. The change was done to support Ivy and dependent builds (see below). The documentation for the new format is available here.

IDE plugins changes

Eclipse plugin with remote run (Subversion) and changes view

link here

TeamCity plugin works with Eclipse 3.2+ and requires plugin Subclipse 1.2.0.

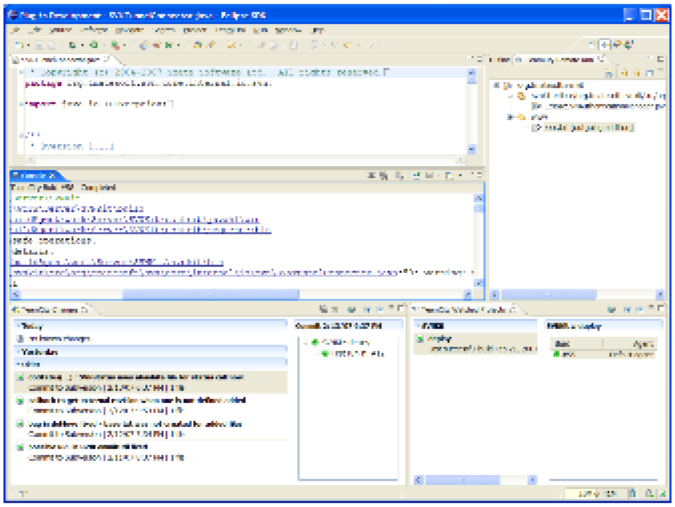
Eclipse plugin for TeamCity features:

- Watched Projects View
  - Displays TeamCity projects user is watching for and their properties.
- Changes View
  - Displays builds triggered by user VCS changes, remote run builds and their properties.

Both watched projects and changes views display TeamCity builds and build configuration statuses, allow to open build log of selected build in Eclipse. Cumulative watched projects and remote run builds statuses are displayed in Eclipse status bar.

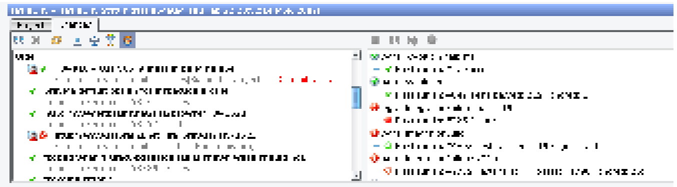
- Remote Run View.
  - Displays locally modified files that could be committed to TeamCity as 'remote run' build. This view is dynamically updated in the same manner as standard Eclipse 'Synchronize' view.

- Navigation
  - Navigation from build logs opened within Eclipse to files referenced in log and from the web client to the Eclipse is supported.
- "Open in Web" actions opens selected objects in web browser.



IntelliJ IDEA plugin

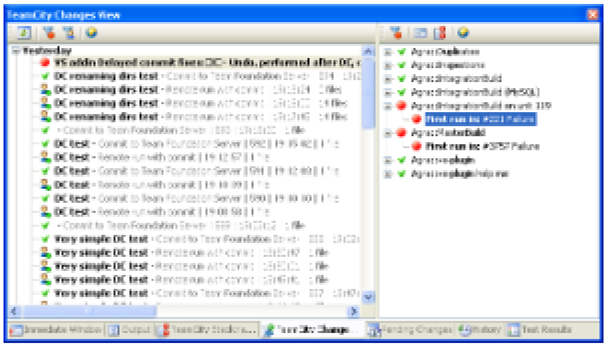
- Option to remember password
- Continuously refresh status of running builds
- Memory leak fixes

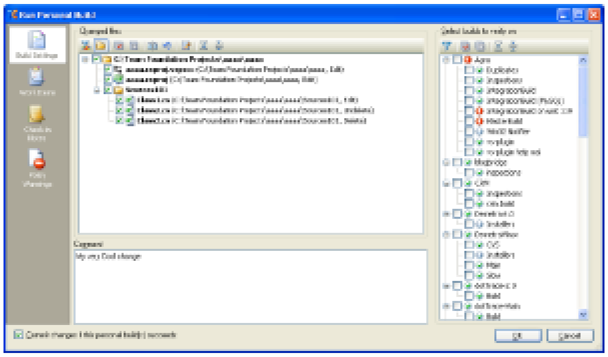


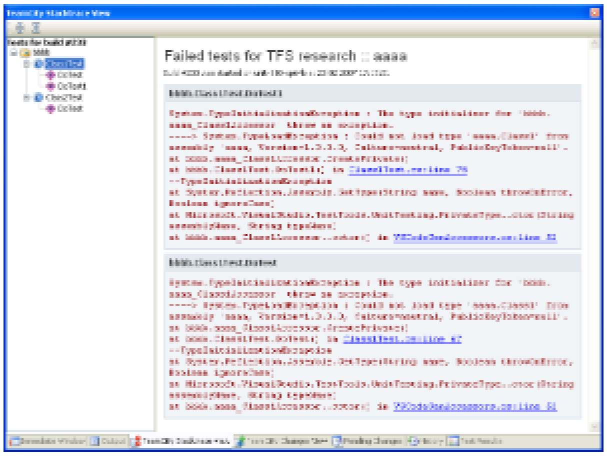
Visual Studio plugin: Changes view + Delayed Commit for TFS

link here

Look at some screenshots:







Failed tests for TFS research :: aaaa

bbbb.Class1Test.DoTest1

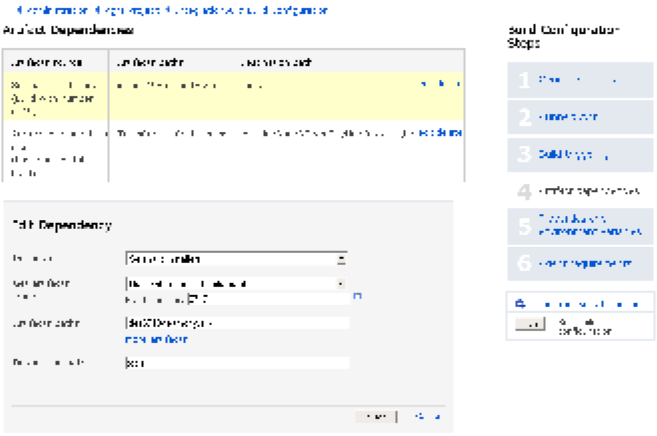
```
System.TypeInitializationException : The type initializer for 'bbbb.
aaaa_Class1Accessor' threw an exception.
----> System.TypeLoadException : Could not load type 'aaaa.Class1' from
assembly 'aaaa, Version=1.0.0.0, Culture=neutral, PublicKeyToken=null'.
at bbbb.aaaa_Class1Accessor.CreatePrivate()
at bbbb.Class1Test.DoTest1() in Class1Test.cs:line 75
--TypeInitializationException
at System.Reflection.Assembly.GetType(String name, Boolean throwOnError,
Boolean ignoreCase)
at Microsoft.VisualStudio.TestTools.UnitTesting.PrivateType..ctor(String
assemblyName, String typeName)
at bbbb.aaaa_Class1Accessor..cctor() in VSCodeGenAccessors.cs:line 31
```

bbbb.Class1Test.DoTest

```
System.TypeInitializationException : The type initializer for 'bbbb.
aaaa_Class1Accessor' threw an exception.
----> System.TypeLoadException : Could not load type 'aaaa.Class1' from
assembly 'aaaa, Version=1.0.0.0, Culture=neutral, PublicKeyToken=null'.
at bbbb.aaaa_Class1Accessor.CreatePrivate()
at bbbb.Class1Test.DoTest() in Class1Test.cs:line 47
--TypeInitializationException
at System.Reflection.Assembly.GetType(String name, Boolean throwOnError,
Boolean ignoreCase)
at Microsoft.VisualStudio.TestTools.UnitTesting.PrivateType..ctor(String
assemblyName, String typeName)
at bbbb.aaaa_Class1Accessor..cctor() in VSCodeGenAccessors.cs:line 31
```

Other changes

Dependent builds and Ivy integration



TeamCity allows to depend one build configuration on another one in terms of artifacts. I.e. if one build configuration requires library produced by another build configuration, you can easily add such dependency and dependent library will be downloaded to the specified place right before build start.
Moreover, TeamCity artifacts repository acts as Ivy repository, so you can reference and use artifacts from TeamCity in your ant builds!

Automatic integration of third-party reporting tools

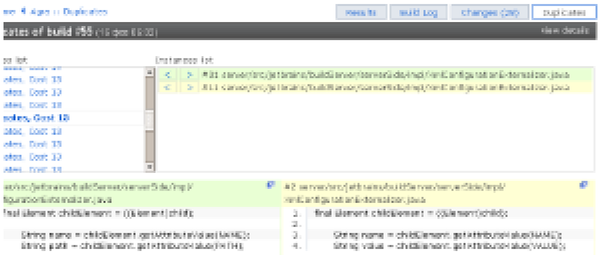
If your build produces some HTML/text/image report which is available as artifact of your build, you can easily create a custom report tab which will show your report embedded right to TeamCity. To do it:

1. In your build script, add a task which archives your whole report into some zip or jar file. Let's name this file "report.zip" (this is step is required if your report contains hundreds of files)
2. For the build configuration, which produces such a report, add "report.zip" (or name of directory with report files) to Artifact paths of this build.
3. In $HOME/.BuildServer/config/main-config.xml add the following line within <server> tag:

```
<report-tab title="A Report" basePath="report.zip" startPage="index.htm" />
<report-tab title="A Report2" basePath="reportDir" startPage="index.html" />
```

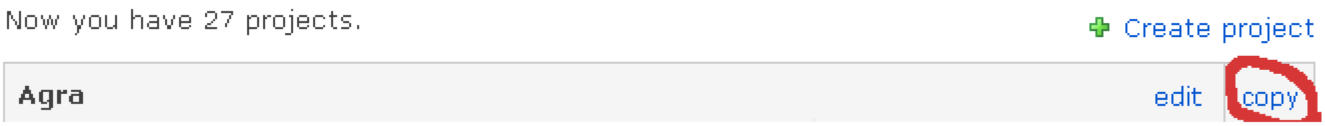
   In this line, "title" is title of the tab and it should be unique, "startPage" specifies relative path of start page within zip file or in directory with report files. Default value for startPage is index.html.

Duplicates runner - Web UI



From now, duplicates results are available in the Web UI. You can easily open file with duplicates in IDE, by clicking the link near file with duplicate.

Possibility to clone projects



Now you have 27 projects.

Enhanced "My Changes" view

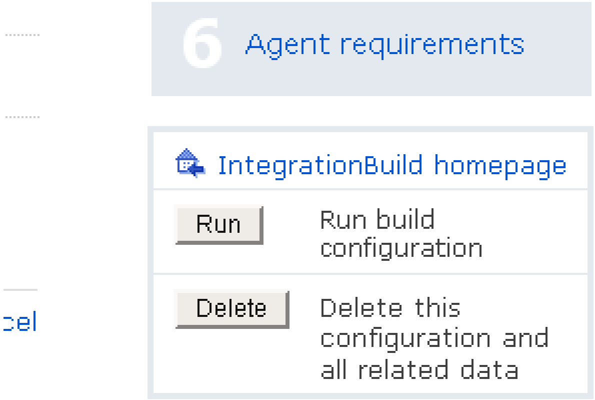
Now you can see the current status of the change, not only the first one. See screenshot.

Option to use current changeset value as build number

Several Maven fixes

- Support Maven modules for tests
- Enable use of altenative pom file (mvn -f mypom.xml)

Add "Run" button to configuration screens



Run build configuration

Delete this configuration and all related data

LDAP authentication

Support pattern-based links to external resources (like Jira issues)

TeamCity data directory

TeamCity data directory can be configured in two ways:

1. Using system property teamcity.data.path passed to TeamCity via JVM option
2. Using environment variable TEAMCITY_DATA_PATH

The default value for TeamCity data directory is $HOME/.BuildServer where $HOME points to user's home directory.