Gaya 8.0 EAP (build 26816) Release Notes

> ⚠ This EAP build does not have license key bundled with it, please use license key from EAP download page

Projects settings directory structure changed

Before TeamCity 8.0 all project configuration files were stored on disk under `<TeamCity Data Directory>/config` with the following structure:

- <TeamCity Data Directory>/config
    - <project name>
        - project-config.xml

Project name was used as the name of the directory with all project settings. Project name was also used as the name of artifacts directory under `<TeamCity Data Directory>/system/artifacts` for all artifacts produced by builds from this project.

In version 8.0 we're changing this structure. Instead of <project name> we'll use project id and all configuration files will be moved under the projects directory:

- <TeamCity Data Directory>/config
    - projects
        - <project id>
            - project-config.xml

Here project id is a new attribute of the project introduced in TeamCity 8.0. The same project id will be used for the directory name under `<TeamCity Data Directory>/system/artifacts`. project id can be assigned from the UI on edit project page. During upgrade TeamCity converter will generate these ids based on project names and rename the directories under `<TeamCity Data Directory>` accordingly.

These new projects ids are also being used in URLs instead of old "projectXXX" ones, as well as in references to the project in other settings files. Old ids (also known as internal ids) are still in use in the database and other non-settings data.
Old web UI URLs are still being handled - users will be redirected to the new ones, so their bookmarks are safe. However, if you use REST API for accessing projects, you will need to change the URLs in the client.

Mixed mode authentication

Before version 8.0, TeamCity server supported the following authentication mechanisms: built-in authentication (enabled by default), Windows Domain (with optional NTLM HTTP authentication), and LDAP. The problem was the authentication modules were mutually exclusive, i.e. it was not possible to have built-in and domain authentication at the same time, although such setup could be convenient during migration from one authentication mode to another. Moreover, if you switched from built-in authentication to domain one, a new set of users was created, thus all the configured user settings, like roles, notification rules and others were lost. Not to mention more advanced cases like the ability to use LDAP connected to Active Directory together with NTLM HTTP (single sign-on scenario).

So we decided to implement a more sophisticated authentication mechanism which we called mixed mode. Starting with this build you can configure several authentication modules, so when the user tries to log in, TeamCity will try all the modules one by one. If one of them authenticates the user, he/she will be logged into TeamCity, if all of them fail to authenticate - he/she won't be able to login. We also implemented a UI to simplify editing of the authentication configuration (in previous versions you had to manually edit main-config.xml file). Finally, we created presets for the most common use cases:

- Default (built-in authentication only)
- LDAP
- LDAP with NTLM
- NTLM

Some modules have settings, these settings can now be edited from the UI too. For advanced users there is an advanced mode

allowing to add / remove authentication modules.

On changing authentication modules, all the existing users are preserved.

Additionally, there is a special superuser account with system administrator role. This account is not editable and it does not have a profile, all you can do is change its username or disable it. When server starts, it generates a password for this account and writes it in teamcity-server.log. This account can be used to recover in case if administrator password was lost.

Related topics in our documentation:

- http://confluence.jetbrains.net/display/TCD8/Configuring+Authentication+Settings
- http://confluence.jetbrains.net/display/TCD8/Super+User

Build history cleanup

We've made several important improvements in the cleanup process:

- server downtime has been reduced by extracting some long cleanup stages to a background process and executing them while the server is running in usual mode, plus several performance improvements have been made in cleanup code
- cleanup progress is shown during cleanup
- it is now possible to specify execution timeout for the cleanup process
- system administrators can stop cleanup from the "cleanup in progress" page
- in previous versions if the server crashed or was killed during cleanup, some files (build logs or artifacts) could be left on disk and never removed. This is not the case anymore, now the server finds and removes such files

Remote debugging of tests on agent (IntelliJ IDEA only)

We already described this feature in our blog post: http://blogs.jetbrains.com/teamcity/2012/12/10/teamhackcity-what-a-bunch-of-teamcity-developers-got-up-to-in-2-weeks/
Now the feature is available in this EAP.

Test status in IDE (IntelliJ IDEA only)

TeamCity IntelliJ IDEA plugin shows status of tests on the TeamCity server right in the editor near the test name:

Statistics

We've made several improvements in statistics charts. First of all, we now use JavaScript charts instead of JFreeChart. These charts work faster, and are more appealing.

A couple of other improvements have been done too:

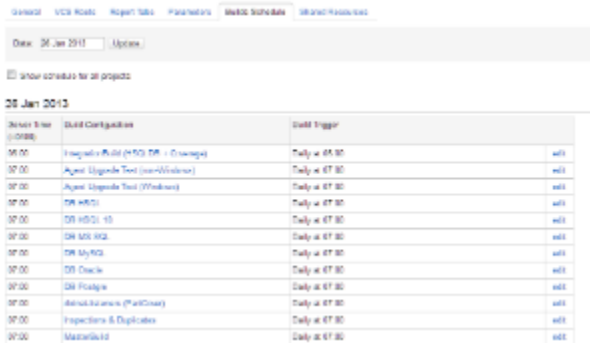- it is now possible to download charts data as CSV file

- options have become available for some charts: Y axis control, whether to use logarithmic scale, min/max value, to show zero or not
- custom color can be set for custom chart series, auto-generation of colors for series has been improved

Shared resources plugin

Recently we published a new plugin which allows to limit the number of running builds based on resources availability. Similar functionality has been part of GroovyPlug for a long time but now it got a user interface and new features.
Read more about the plugin here.

Builds schedule tab

Builds schedule tab on the project level shows a schedule of builds triggering based on information in schedule triggers.
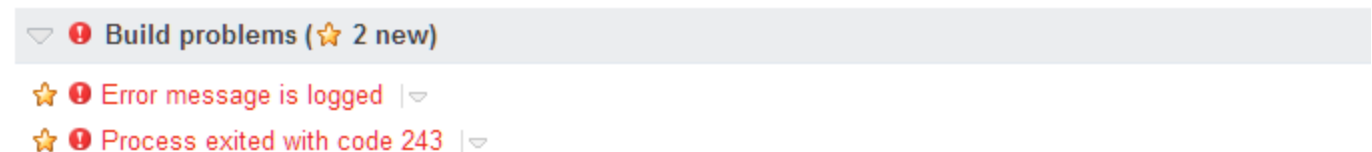

IntelliJ IDEA Inspections

- IntelliJ IDEA inspections runner now supports projects with Android SDK
- IntelliJ IDEA indexes are preserved on agent and reused between inspections builds on the same agent
- it is now possible to configure specific set of plugins to be loaded by IntelliJ IDEA on agent (see "disabled plugins" setting on edit build step page)

Build problems

We continued improving features related to build problems. In version 7.1 we introduced the concept of a build problem mainly to make it clear why the build failed.
Now we improved it and TeamCity is able to detect whether the problem is a new one in this build or not.
Next to come - abilities to mute problems or assign investigations for them.

Other changes

- Change log filter now allows to filter changes by comment. Some facelifting has been applied to filters on various pages.
- System administrator can now pause the entire build queue from the build queue page, in this case no builds will start until the queue is resumed.
- Mercurial plugin now works with bookmarks in the same way it works with branches, in other words it is now possible to use Mercurial bookmarks as feature branches in TeamCity
- Build progress bar, which you could previously see near running builds on build configuration page was added to the running build overview page as well
- Build compilation errors can now be quickly viewed in your IDE via "open in IDE" functionality (only in IntelliJ IDEA and Eclipse so far)
- Visual diff between "actual" and "expected" values in tests failed by comparison failure now can be seen right on the web
- Searching of suitable configurations for remote run in IntelliJ IDEA plugin was sped up; moreover, now you can skip this

step altogether and see all available build configurations instantly (suitable build configurations will continue to load in the background)
- VCS status tab was added to the Diagnostics page, this tab shows activity of "checking for changes" tasks and the time it took to collect changes in different VCS roots
- internal database engine (HSQL DB) has been upgraded to version 2.0 (affects new installations only)
- Test reordering is supported for TestNG 6.x
- fixed issues (http://youtrack.jetbrains.com/releaseNotes/TW?q=%23fixed+Fix+versions%3A+\{Gaya+8.0+EAP+%2826816%29\}+Fix+versions%3A+-\{Faradi+7.1.1+%2824074%29\}+Fix+versions%3A+-\{Faradi+7.1.2+%2824170%29\}+Fix+versions%3A+-\{Faradi+7.1.3+%2824266%29\}+-\{trunk+issue\}+-Task&title=Gaya+8.0+EAP+%2826816%29&showDescription=false&showComments=false)