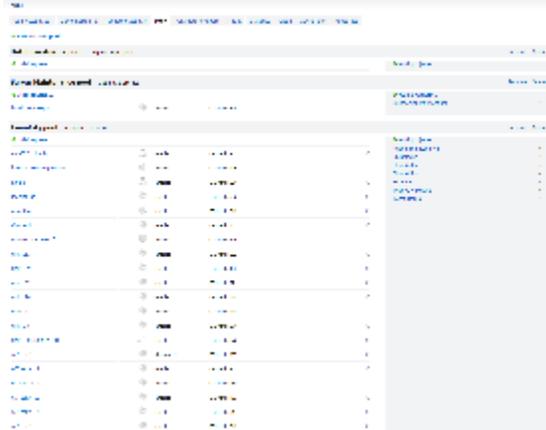


Faradi 7.0 EAP (build 20334) Release Notes

- New agent pools administration interface
- .NET Inspections runner (based on R#)
- Global Maven settings
- Fail build on a specific text in build log
- Per-check-in builds
- Graph of commits on the build configuration changelog page
- Reworked snapshot dependencies graph
- Types for build properties (experimental)
- Filtering messages in build log tree view
- Other improvements

New agent pools administration interface

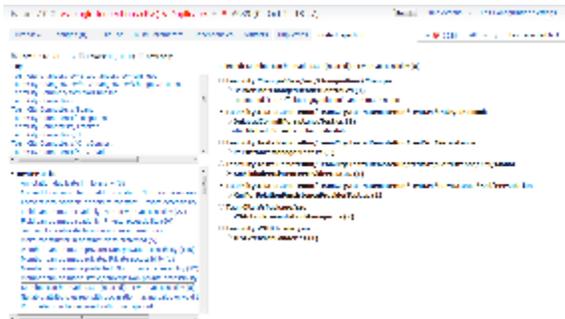
We reworked agent pools administration interface, and we would appreciate any feedback on it.



.NET Inspections runner (based on R#)

New .NET Inspections runner based on ReSharper inspections engine is now available. .NET duplicates runner has got major update too.

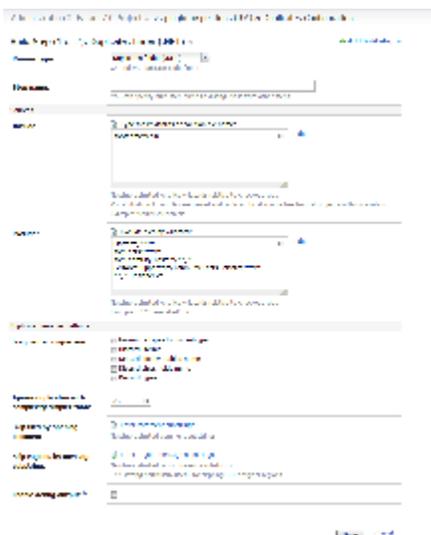
.NET Inspections runner results:



Inspections runner configuration screen:



New duplicates runner configuration screen:



Global Maven settings

Now it is possible to define global (per server) settings xml files, and use them in your Maven builds.

Global settings xml files can be uploaded via TeamCity administration interface:



The files are stored under <TeamCity Data Directory>/config/_mavenSettings directory. If necessary, the files can be edited right there.

Then in each Maven build step you can specify which settings xml to use: default (chosen by Maven), specified by path or global (uploaded on server).

BUILD Step + Add build step

Builder type: Maven Runner for Maven builds

Step name:

Goals: Passes or fails build depending on goal to include.

Fail to a build file: Used to fail build to the checkout directory.

Source folder path: Optional parameter, needed when you are not going to use Maven build with a source folder. Run test step first.

Additional Maven command line parameters: The following parameters are separated by a space. Run test step first.

Working directory: Optional, specify a different build directory.

User settings: Path to a Maven settings file.

User settings path: Path to a user settings file.

Proxied settings: URL of a proxy server. Run test step first.

Java Parameters

Java home path: Path to a Java home directory. Run test step first.

Java command line parameters:

Incremental build:

Enable incremental build: Build only modules affected by changes.

Code coverage:

Choose coverage system:

Fail build on a specific text in build log

Build failure condition to fail build on a specific text in build log has been improved. You can read more about it in [this blog post](#).

Per-check-in builds

VCS trigger now supports two new options:

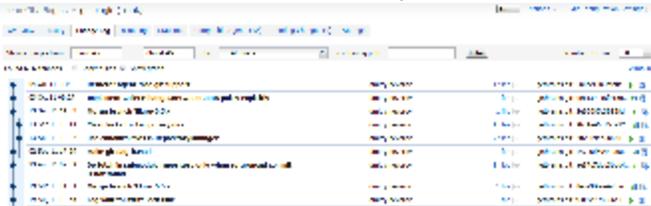
- Trigger one build per each checkin
- Trigger one build per group of checkins from the same user

With the first option enabled, each new check-in will result in a build added to queue and associated with this check-in (i.e. build won't include other changes).

If second option is enabled, and there are several pending changes, TeamCity will group them by user and will start builds having single user changes only.

Graph of commits on the build configuration changelog page

If your project uses Git or Mercurial you can see graph of commits on build configuration change log page. Graphs are also useful for non-DAG-based VCSes, they make it easier to understand where a VCS root modification comes from.



Reworked snapshot dependencies graph

Build dependencies tab has been reworked to show complete build chain formed by snapshot dependencies. The same view is also available on change details page.



Types for build properties (experimental)

Disclaimer: This feature is experimental and will be changed in the future releases. No data conversions will occur and you will need to re-enter all property descriptions again.

With this build property definitions of a build configurations get an additional "Spec" attribute. This property "meta" information is then used to display the property in the "Run custom build" dialog. It can be used to:

- specify property presentation (checkbox, drop-down, validation for valid integer value)
- specify display text to use instead of property name
- mark property as mandatory (user needs to specify it's value before running a build)
- mark property as hidden (the property will not appear in custom run dialog)

These settings are configured via a specially-formatted string described in the [comment](#) of the original issue.

Some examples are:

```
text label='My pretty and important name' required='true'  
text hidden='true'  
integer label='Number of retries' minValue='0' maxValue='10'  
checkbox  
enum items='red;green;blue'  
enum label='JVM Memory' items='small;medium;large' data_small='-Xmx100m' data_medium='-Xmx300m'  
data_large='-Xmx1000m'
```

Filtering messages in build log tree view

We continue improving tree view in build log, and with this EAP we added filter by message status, and also improved tree view presentation.

Other improvements

- global, per-server build execution timeout, can be overridden on build configuration level
- Performance Monitor is now a build feature and is not enabled by default (you'll need to add it explicitly), also it now supports MacOS X
- NuGet integration improved
- On My Changes page and on the page for an individual Change, you can see an estimate for queued builds associated with the change (see Builds tab)
- [full list of fixed issues](#)